(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) **EP 1 970 830 B1**

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
17.08.2011 Bulletin 2011/33

(51) Int Cl.:
G06F 21/00 *(2006.01)*

(21) Application number: **08152627.9**

(22) Date of filing: **12.03.2008**

(54) **Information processing apparatus, software update method, and image processing apparatus**

Informationsverarbeitungsgerät, Softwareaktualisierungsverfahren und Bildverarbeitungsverfahren

Appareil de traitement d'informations, procédé de mise à jour logiciel, et appareil de traitement d'image

(84) Designated Contracting States:
DE ES FR GB IT NL

(30) Priority: **15.03.2007 JP 2007067250**

(43) Date of publication of application:
17.09.2008 Bulletin 2008/38

(73) Proprietor: **Ricoh Company, Ltd.**
Tokyo 143-8555 (JP)

(72) Inventor: **Okabe, Kiwamu**
Yokohama-shi
Kanagawa (JP)

(74) Representative: **Schwabe - Sandmair - Marx**
Patentanwälte
Stuntzstraße 16
81677 München (DE)

(56) References cited:
WO-A-99/39475
US-A1- 2003 194 094
US-B1- 6 185 678

EP 1 970 830 B1

Printed by Jouve, 75001 PARIS (FR)

Description

BACKGROUND OF THE INVENTION

1. Field of the Invention

[0001] The present invention relates to an information processing apparatus, a software update method, and an image processing apparatus, and more specifically to an information processing apparatus or an image processing apparatus having a primary module and a backup module, and a software update method of the information processing apparatus or the image processing apparatus.

2. Description of the Related Art

[0002] As security becomes increasingly critical, information processing apparatuses such as personal computers and image processing apparatuses such as Multi Function Peripherals (MFP) capable of encrypting information stored in the apparatuses to avoid wiretapping have become available lately. For example, Patent Document 1 describes a PC adopting the specifications of Trusted Computing Platform Alliance (TCPA) in which information is encrypted using a Trusted Platform Module (TPM). The TPM is realized in a chip directly mounted on, for example, a motherboard.

[0003] On the other hand, to respond to a failure, for example, a duplexing system has been employed in information processing apparatuses such as personal computers and image processing apparatuses such as MFPs. Furthermore, to respond to a bug, a security hole, addition or modification of functions, the programs have also been updated in information processing apparatuses such as personal computers and image processing apparatuses such as MFPs (see, for example, Patent Document 2).

[0004] Document US 6,185,678 B1 discloses an information processing apparatus including primary and backup modules necessary to boot the apparatus. Further disclosed is the computation of a cryptographic hash value of a module and comparing this value with a stored digital signature.

[0005] Document US 2003/194094 A1 discloses an operating system (DRMOS) which encrypts a seed and access predicates using an OS storage key. The seed is used as an operand for a subsequent password generation function while the access predicates contain criteria which have to be met by an application in order to the operation system granting the application access to the seed.

[0006] Herein, a conventional method of encrypting and decrypting information using the TPM, and a program update (hereinafter referred to as "ROM update") are briefly described. FIG. 1 shows an exemplary configuration of a conventional information processing apparatus. The information processing apparatus includes a CPU 1, a BIOS ROM 2, a disk 3, a non-volatile (NV) RAM 4, and a main memory 5 as the hardware configuration. The CPU 1, the BIOS ROM 2, the disk 3, the NVRAM 4, and the main memory 5 are connected to each other via a bus 6.

[0007] The BIOS ROM 2 stores a Basic Input/Output System (BIOS) 10 module. The disk 3 stores a loader 11, a kernel 12, and a root file system (Rootfs) 13 modules. The NVRAM 4 stores plain text data 14 that users use.

[0008] The root file system 13 manages a boot program 21, a ROM update flag control program 22, a blob decryption section 23, and an application 24 that are stored in the disk 3. It should be noted that each of the BIOS 10, the loader 11, the kernel 12, the root file system 13 modules and the like is loaded into the main memory 5 to be executed. In the following, the BIOS 10, the loader 11, the kernel 12, the root file system 13 modules and the like are described as processing subjects.

[0009] A boot sequence of the information processing apparatus in FIG. 1 is described with reference to FIG. 2. FIG. 2 is a sequence diagram showing the processes of the information processing apparatus being booted. In step S1, the BIOS 10 loads and boots the loader 11. In steps S3 through S5, the loader 11 loads and boots the kernel 12 and the root file system 13.

[0010] In step S6, the kernel 12 boots the boot program 21 in the root file system 13. In step S7, the boot program 21 boots the application 24 in the root file system 13. In step S8, the application 24 is now capable of writing data into the NVRAM 4 and reading, for example, plain data 14 in the NVRAM 4.

[0011] Next, a mechanism of the TPM is briefly described. In the following, an example where the loader 11 boots the kernel 12 is described.

[0012] FIG. 3 is a diagram schematically showing a process of storing a hash value into the TPM 7. In step S11, the loader 11 loads the kernel 12 from the disk 3 into the main memory 5. In step S12, the TPM 7 stores, for example, a hash value into a Platform Configuration Register (PCR), the hash value being calculated based on a method of generating a fixed-length pseudo random number from an original document. In FIG. 3, a hash value "0x3a" is stored in a "PCR3". In step S13, the loader 11 boots the kernel 12.

[0013] In this manner, when the TPM 7 boots, for example, the BIOS 10, the loader 11, the kernel 12 and the root file system 13 modules, the TPM 7 stores hash values calculated from the modules in the PCRs.

[0014] FIG. 4 is a drawing schematically showing a decrypting process of the information using the TPM 7. In the TPM 7, four hash values calculated from the corresponding modules are stored in "PCR1" through "PCR4". When the information is decrypted using the TPM 7, a Blob A 41 and a Blob B 42 each including at least one of the "PCR1" through "PCR4" data are used.

[0015] In the Blob A 41, a value "0x3a" is stored in the "PCR3". In the Blob B 42, values "0xe9"', "0x12", "0x3b",

and "0x06" are stored in the "PCR1" through the "PCR4", respectively. In the TPM7, values "0xe9", "0x12", "0x3a", and "0x06" are stored in its "PCR1" through the "PCR4", respectively.

[0016] In case of Blob A 41, the same hash value is in the "PCR3" of the Blob A 41 and the "PCR3" of the TPM 7. Therefore, the TPM 7 permits taking the information from the Blob A 41. In case of Blob B 42, a hash value in the "PCR3" of the Blob A 41 is different from that in the "PCR3" of the TPM 7. Therefore, the TPM 7 does not permit taking the information from the Blob A 41. It should be noted that when "no setting" may be stored in, for example, the "PCR1", the "PCR2", and the "PCR4" in the Blob A 41, the TPM 7 does not use the register to determine whether to permit taking the information.

[0017] FIG. 5 shows an exemplary configuration of an information processing apparatus having the TPM. The information processing apparatus in FIG. 5 includes the CPU 1, the BIOS ROM 2, the disk 3, the NVRAM 4, the main memory 5, the TPM 7, and a Hard Disk Drive (HDD) 8 as the hardware configuration. The CPU 1, the BIOS ROM 2, the disk 3, the NVRAM 4, the main memory 5, the TPM 7, and a Hard Disk Drive (HDD) 8 are connected to each other via a bus 6.

[0018] The configuration of the information processing apparatus in FIG. 5 is different from that in FIG. 1 in that the information processing apparatus in FIG. 5 further includes the TPM 7 and the HDD 8. Furthermore, the disk 3 stores a Blob 43 in addition to the configuration in FIG. 1. The Blob 43 includes an encrypted encryption key 51 for the NVRAM 4. The Blob 43 stores hash values each calculated from the BIOS 10, the loader 11, the kernel 12, and the root file system 13 in the "PCR1" through "PCR4", respectively.

[0019] The NVRAM 4 stores encrypted data 15 in addition to the plain text data 14. The HDD 8 stores encrypted data 16. The same reference numerals are used in the figure to describe those components that are identical to the components of FIG .1 without repeated description. The description of the Blob having an encrypted encryption key of the HDD 8 is also omitted.

[0020] A boot sequence of the information processing apparatus in FIG. 5 is described with reference to FIG. 6. FIG. 6 is a sequence diagram showing exemplary processes of the information processing apparatus being booted. In step S21, the BIOS 10 loads the loader 11. In step S22, a hash value of the loader 11 is stored in a PCR of the TPM 7. In step S23, the BIOS 10 boots the loader 11.

[0021] In step S24, the loader 11 loads the kernel 12. In step S25, a hash value of the kernel 12 is stored in a PCR of the TPM 7. In step S26, the loader 11 loads the root file system 13. In step 27, a hash value of the root file system is stored in a PCR of the TPM 7.

[0022] In step S28, the loader 11 boots the kernel 12 and the root file system 13. In step S29, the kernel 12 boots the boot program 21 in the root file system 13. In steps 30 and 31, the boot program 21 boots the blob decryption section 23 and the application 24 in the root file system 13.

[0023] In step S32, the blob decryption section 23 acquires the encryption key 51 for the NVRAM 4 from inside the Blob 43. In step S33 by using the encryption key, the application is now capable of writing encrypted data into the NVRAM 4 and reading encrypted data 14 stored in the NVRAM 4.

[0024] Patent Document 1: Japanese Patent Application Publication No. 2004-282391

[0025] Patent Document 2: Japanese Patent Application Publication No. 2005-196745

[0026] However, in an information processing apparatus having a configuration as shown in FIG. 5, the following problem may occur during the ROM update. FIG. 7 is a drawing schematically illustrating a problem having occurred during the ROM update. In an information processing apparatus having a configuration as shown in FIG. 5, when the BIOS 10 stored in the BIOS ROM 2 is replaced by a new BIOS 10a, the Blob 43 corresponding to the BIOS 10 is required to be updated to a Blob A 43a that corresponds to the BIOS 10a.

[0027] Unfortunately, in a conventional information processing apparatus, when an update process from the Blob A 43 to the Blob A 43a is interrupted due to some reason, the hash value stored in the "PCR1" of the TPM 7 may become different from the hash value stored in the "PCR1" of the Blob A 43a. A problem arises that when the hash value stored in the "PCR1" of the TPM 7 becomes different from the hash value stored in the "PCR1" of the Blob A 43a, in that the encryption key 51 for the NVRAM 4 cannot be taken from the Blob 43a, resulting in that the encrypted data stored in the NVRAM 4 cannot be decrypted.

[0028] This problem illustrated in FIG. 7 can be solved when an information processing system has a configuration as shown in FIG. 8. The information processing apparatus in FIG. 8 includes a primary system 81 and a backup system 82, constituting a duplex system. The primary system 81 includes the BIOS 10, the loader 11, the kernel 12, and the root file system 13. The backup system 82 includes the BIOS 10b, a loader 11b, a kernel 12b, and a root file system 13b.

[0029] It should be noted that the BIOS 10, the loader 11, the kernel 12, and the root file system 13 are included in primary modules, and the BIOS 10b, the loader 11b, the kernel 12b, and the root file system 13b are included in backup modules.

[0030] Typically, an information processing apparatus is booted sequentially in an order of the BIOS 10, the loader 11, the kernel 12, and the root file system 13. Hereinafter, a procedure of booting like this is referred to as a "boot path". In the example of FIG. 8, due to an error having occurred in the loader 11, the booth path becomes: BIOS 10 → loader 11b → kernel 12 → root file system 13.

[0031] That is, in an information processing apparatus having the backup system 82, when a module of the pri-

mary system has a problem, the same kind of module in the backup system 82 can usually be booted. A booth path can be changed by, for example, a ROM update flag control program.

[0032] Because of this structure, there is a problem that the same number of Blobs which is equal to the number of booth paths defined by the combination of the modules in the primary system 81 and the modules in the backup system 82 are required to be provide. FIG. 9 is a drawing schematically illustrating a problem that may occur when information is encrypted and decrypted using the TPM in an information processing apparatus having a backup system.

[0033] Further, there is another problem in an information processing apparatus having a configuration as shown in FIG. 9 that when the BIOS 10 stored in the BIOS ROM 2 is updated to the BIOS 10a, all of the plural Blobs corresponding to the BIOS 10 are required to be updated so as to correspond to the BIOS 10a. FIG. 10 is a drawing schematically illustrating a problem occurring while information is encrypted and decrypted using the TPM, where the ROM update is executed in an information processing apparatus having a backup system.

[0034] As described, when a conventional system is arranged to employ a duplex system having both a primary system and a backup system, have a ROM update capability, and improve the security by adding both an encryption and a decryption capability of information by using the TPM 7, it takes a lot of effort to manage the Blobs 73.

SUMMARY OF THE INVENTION

[0035] The present invention is made in light of the problems and may provide an information processing apparatus, a method of software update, and an image processing apparatus capable of encrypting and decrypting information using values uniquely calculated from a booted primary module and a booted backup module with much easiness.

[0036] To solve the problems, according to a first aspect of the present invention, there is provided an information processing apparatus including one or more kinds of primary modules necessary to boot the apparatus and one or more kinds of backup modules to be used when the primary modules fail, so that the information processing apparatus is booted in a manner that when any kind of the primary modules fails, the same kind of backup module is used. The information processing apparatus includes a value storage unit storing values uniquely calculated from the one or more kinds of the primary modules or the backup modules used when the apparatus is booted, an encryption information storage unit storing information unique to the each kind of the primary or the backup modules, the information being encrypted based on a value calculated from the each kind of the primary modules or the backup modules, an information decryption unit decrypting the information unique to the each kind of the primary modules or the backup modules using the values in the value storage unit, the information being stored in the encryption information storage unit, and an encryption information update unit, when any of the primary modules or the backup modules is updated, encrypting the information unique to the each kind of the primary modules or the backup modules based on a value calculated from the each kind of the primary modules or the backup modules after the update, the information being stored in the encryption information storage unit.

[0037] Further, to solve the problems, according to a second aspect of the present invention, there is provided an image processing apparatus including one or more kinds of primary modules necessary to boot the apparatus, one or more kinds of backup modules to be used when the primary modules fail, a plotter section and scanner section so that the plotter and the scanner sections are booted in a manner that when any kind of the primary modules fails, the same kind of backup module is used. The image processing apparatus includes a value storage unit storing values uniquely calculated from the one or more kinds of the primary modules or the backup modules used when the apparatus is booted, an encryption information storage unit storing information unique to the each kind of the primary or the backup modules, the information being encrypted based on a value calculated from the each kind of the primary modules or the backup modules, an information decryption unit decrypting the information unique to the each kind of the primary modules or the backup modules using the values in the value storage unit, the information being stored in the encryption information storage unit, and an encryption information update unit, when any of the primary modules or the backup modules is updated, encrypting the information unique to the each kind of the primary modules or the backup modules based on a value calculated from the each kind of the primary modules or the backup modules after the update, the information being stored in the encryption information storage unit.

[0038] It should be noted that any method, apparatus, system, computer program, recording medium, data structure including a constitutional element, an expression, or any combination of the present invention may be included in embodiments of the present invention.

[0039] According to an embodiment of the present invention, there is provided an information processing apparatus, a method of software update, and an image processing apparatus capable of encrypting and decrypting information using values uniquely calculated from a booted primary module and a booted backup module with less efforts.

BRIEF DESCRIPTION OF THE DRAWINGS

[0040] Other objects, features, and advantages of the present invention will be apparent from the following detailed description when read in conjunction with the accompanying drawings, in which:

FIG. 1 is a drawing showing an exemplary configuration of a conventional information processing apparatus;
FIG. 2 is a sequence diagram showing an exemplary process of the information processing apparatus when the information processing apparatus is booted;
FIG. 3 is a drawing showing a process of storing a hash value in the TPM;
FIG. 4 is a drawing schematically showing a process of decrypting information using the TPM;
FIG. 5 is a drawing showing an exemplary configuration of a conventional information processing apparatus including the TPM;
FIG. 6 is a sequence diagram showing an exemplary process of the information processing apparatus when the information processing apparatus is booted;
FIG. 7 is a drawing schematically showing a problem having occurred during a ROM update process;
FIG. 8 is a drawing schematically showing a duplex system including a primary system and a backup system;
FIG. 9 is a drawing schematically showing a problem occurring when information is encrypted and decrypted using the TPM in a conventional information processing apparatus including the TPM;
FIG. 10 is a drawing schematically showing a problem occurring when ROM is updated while information is encrypted and decrypted using the TPM in a conventional information processing apparatus including the TPM;
FIG. 11 is a drawing showing an exemplary configuration of an information processing apparatus according to an embodiment of the present invention;
FIG. 12 is a drawing showing an exemplary module configuration of the disk of the information processing apparatus;
FIG. 13 is a drawing showing an exemplary module configuration in a root file system in the disk;
FIG. 14 is a sequence diagram showing a process of the information processing apparatus when the information processing apparatus is booted;
FIG. 15 is a diagram schematically showing a process of a ROM update;
FIG. 16 is a drawing schematically showing a process of an encryption key update;
FIG. 17 is a drawing showing another module configuration of the disk in the information processing apparatus;
FIG. 18 is a drawing showing an exemplary module configuration of the disk in the information processing apparatus according to an embodiment of the present invention;
FIG. 19 is a drawing showing information stored in the NVRAM of the information processing apparatus;
FIG. 20 is a drawing showing information stored in the HDD of the information processing apparatus; and
FIGS. 21 and 22 are drawings schematically showing a process of decrypting encrypted information in the NVRAM when the disk has crashed.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0041] Next, best modes for carrying out the invention are described with reference to exemplary embodiments of the present invention and accompanying drawings. In the embodiments, an information processing apparatus such as a personal computer is described. However, the embodiment is not limited to such an information processing apparatus, and may be carried out in, for example, an image processing apparatus such as a Multi Function Peripheral (MFP).

[Embodiment 1]

[0042] FIG. 11 shows an exemplary configuration of an information processing apparatus according to an embodiment of the present invention. The information processing apparatus in FIG. 11 includes a CPU 1, a BIOS ROM 2, a disk 3, an NVRAM 4, a main memory 5, a TPM 7, and a HDD 8 as a hardware configuration. The CPU 1, the BIOS ROM 2, the disk 3, the NVRAM 4, the main memory 5, the TPM 7, and the HDD 8 are connected to each other via the bus 6.

[0043] The BIOS ROM 2 includes a BIOS 10 as a primary module and a BIOS 10b as a backup module. The NVRAM 4 stores plain text data 14 and encrypted data 15 that a user uses. The HDD 8 stores encrypted data 16.

[0044] FIG. 12 shows an exemplary configuration of modules stored in the disk 3. In FIG. 12, the disk 3 includes a loader 11, a kernel 12, and a root file system 13 as primary modules; a loader 11b, a kernel 12b, and a root file system 13b as backup modules; Blobs 60a through 60h; and an encrypted encryption key 62 for the NVRAM 4.

[0045] The Blob 60a includes an encrypted key "A". The Blob 60b includes an encrypted key "B". The Blob 60c includes an encrypted key "C". The Blob 60d includes an encrypted key "D". The Blob 60e includes an encrypted key "A". The Blob 60f includes an encrypted key "B". The Blob 60g includes an encrypted key "C". The Blob 60h includes an encrypted key "D".

[0046] As a result, the Blob 60a and the Blob 60e have the same key "A", the Blob 60b and the Blob 60f have the same key "B", the Blob 60c and the Blob 60g have the same key "C", and the Blob 60d and the Blob 60h have the same key "D".

[0047] Further, hash values calculated based on a calculation method of generating a fixed-length pseudo random number from the BIOS 10 and 10b are stored into each "PCR1" of the Blobs 60a and 60e, respectively. In the same manner, the hash values calculated from the

loader 11 and 11b are stored into each "PCR2" of the Blobs 60b and 60f, respectively. The hash values calculated from the kernel 12 and 12b are stored into each "PCR3" of the Blobs 60c and 60g, respectively. The hash values calculated from the root file system 13 and 13b are stored into each "PCR4" of the Blobs 60d and 60h, respectively.

[0048] With the configuration where Blobs 60a through 60h are provided as shown in FIG. 12, the keys A through D can be obtained when either primary modules or backup modules are booted in the boot path. Further, in the configuration where eight Blobs 60a through 60h are provided as shown in FIG. 12, the keys A through D can be obtained in any possible boot. The encryption key 62 for the NVRAM 4 is encrypted using the keys "A" through "D".

[0049] FIG. 13 shows an exemplary module configuration of the root files systems 13 and 13b in FIG. 12. The module configuration of the root file systems 13 and the module configuration of the root file systems 13b are identical. An explanation of the root files systems 13 is described below, and an explanation of the root files systems 13b is omitted.

[0050] The root file system 13 manages a boot program 21, a ROM update flag control program 22, a blob decryption section 23, an application 24, a blob update program 25, and an encryption key update program 26 that are stored in the disk 3.

[0051] The boot program 21 boots the application 24 in the root file system 13. The ROM update flag control program 22 controls the boot path defining a boot flow. The blob decryption section 23 acquires the keys "A" through "D" from the Blobs 60a through 60h using the TPM 7. The blob update program 25 controls the update of the Blobs 60a through 60h. The encryption key update program 26 controls the update of the encryption key 62 for the NVRAM 4.

[0052] Referring back to FIG. 11, when each kind of the primary or backup modules of the BIOS 10 or the BIOS 10b, the loader 11 or the loader 11b, the kernel 12 or the kernel 12b, and the root file system 13 or the root file system 13b is booted, the hash values of the modules used for the boot are accordingly stored in the "PCR1" through "PCR4", respectively, of the TPM 7.

[0053] That is, the hash value calculated from the BIOS 10 or 10b is stored in the "PCR1" of the TPM 7; the hash value calculated from the loader 11 or 11b is stored in the "PCR2" of the TPM 7; the hash value calculated from the kernel 12 or 12b is stored in the "PCR3" of the TPM 7; and the hash value calculated from the root file system 13 or 13b is stored in the "PCR4" of the TPM 7.

[0054] The modules including the BIOS 10 and 10b, the loader 11 and 11b, the kernel 12 and 12b, and the root file system 13 and 13b are loaded into the main memory by the CPU 1 and executed. In the following descriptions, the modules including the BIOS 10 and 10b, the loader 11 and 11b, the kernel 12 and 12b, and the root file system 13 and 13b are described as processing subjects, for explanation purposes.

[0055] Next, a boot sequence of the information processing apparatus in FIG. 11 is described with reference to FIG. 14. FIG. 14 is a sequence diagram showing an exemplary booting process of the information processing apparatus according to an embodiment of the present invention. It is assumed that the hash value of the BIOS 10 is already stored in the "PCR1" of the TPM7 before step S41 in FIG. 14.

[0056] In step S41, BIOS loads the loader 11. In step S42, the hash value of the loader 11 is stored in the "PCR2" of the TPM 7. In step S43, the BIOS boots the loader 11.

[0057] In step S44, the loader 11 loads the kernel 12. In step S45, the hash value of the kernel 12 is stored in the "PCR3" of the TPM 7. In step S46, the loader loads the root file system 13. In step S47, the hash value of the root file system 13 is stored in the "PCR4" of the TPM 7.

[0058] In step S48, the loader 11 boots the kernel 12 and the root file system 13. In step S49, the kernel 12 boots the boot program 21 in the root file system 13. In steps S50 and S51, the boot program 21 boots the blob decryption section 23 and the application 24, respectively, in the root file system 13.

[0059] In step S52, the blob decryption section 23 acquires the keys "A" through "D" from the blobs 60a through 60d, respectively, using the TPM 7. In step S53, the blob decryption section 23 decrypts the encrypted encryption key 62 for the NVRAM 4 using the acquired keys "A" through "D". In step S54, the application is now capable of writing encrypted data 15 into the NVRAM 4 and reading encrypted data in the NVRAM 4 using the decrypted encryption key 62.

[0060] In the following, a specific process of ROM update and encryption key update in the information processing apparatus in FIG. 11 is described.

(ROM UPDATE)

[0061] FIG. 15 is a diagram schematically showing a process of the ROM update. FIG. 15 shows an example where the kernel 12 is updated to a new kernel 12a. First, the ROM update flag control program 22 changes the boot path indicating the procedure of the boot from BIOS 10 → loader 11 → kernel 12 → root file system 13 to BIOS 10 → loader 11 → kernel 12b → root file system 13. Then by rebooting the information processing apparatus, the BIOS 10, the loader 11, the kernel 12b, and the root file system 13 are booted accordingly.

[0062] In step S61, the kernel 12 is replaced by the new kernel 12b. In step S62, the blob decryption section 23 acquires the key "C" from the Blob 60c using the TPM 7 in the same manner as described above.

[0063] In step S63, the blob update program 25 generates a hash value calculated from the new kernel 12a. In step S64, the blob update program 25 generates a new Blob 60i including the generated hash value. In step S65, the blob update program 25 replaces the Blob 60c by the generated Blob 60i. Then, the ROM update flag

control program 22 restores the boot path to BIOS 10 → loader 11 → kernel 12 → root file system 13.

[0064] During the process of the ROM update shown in FIG. 15, even when the update process from the Blob 60c to the Blob 60i is interrupted for some reason, since the same key "C" is stored in the Blob 60g, the key "C" can be acquired from the Blob 60g. As a result, the encrypted encryption key 62 for the NVRAM 4 can be decrypted by using the acquired keys "A" through "D", and accordingly, the encrypted data in the NVRAM 4 can be decrypted.

(Encryption key update)

[0065] FIG. 16 is a diagram schematically showing a process of the encryption key update. In step S71, the encryption key update program 26 creates a backup copy of the encrypted data 15 in the NVRAM 4 and stores the created backup copy in the disk 3. In step S72, the blob decryption section 23 acquires the keys "A" through "D" from the Blobs 60a through 60d, respectively, corresponding to the boot path. In step S73, the encryption key update program 26 encrypts an encryption key 62a for the NVRAM 4 using the acquired keys "A" through "D", and stores the encrypted encryption key 62a in the disk 3.

[0066] In step S74, the blob decryption section 23 decrypts the encrypted encryption key 62 using the keys "A" through "D", and acquires the decrypted encryption key 62. In step S75, the encryption key update program 26 decrypts the encrypted data 15 stored in the NVRAM 4 using the decrypted encryption key 62. In step S76, the encryption key update program 26 encrypts the decrypted encrypted data 15 again using the new encryption key 62a for the NVRAM 4.

[0067] In step S77, the encryption key update program 26 deletes the encrypted data 15 stored in the disk 3 as a backup copy in step S71. In step S78, the encryption key update program 26 further deletes the encrypted encryption key 62 for the NVRAM 4 stored in the disk 3.

[0068] During the above process of the encryption key update shown in FIG. 16, even when the update from the encryption key 62 to the new encryption key 62a is interrupted, since the copy of the encrypted data 15 is stored in the disk 3 as a backup, it is possible to perform the process of the encryption key update again.

(Another configuration of the disk 3)

[0069] FIG. 17 shows another exemplary module configuration in the disk 3. The disk 3 in FIG. 17 includes primary modules of the loader 11, the kernel 12, and the root file system 13; backup modules of the loader 11b, the kernel 12b, the root file system 13b; and Blobs 60a through 60c, 60e through 60g, 60j, and 60k.

[0070] The Blob 60a includes the encrypted key "A". The Blob 60b includes the encrypted key "B". The Blob 60c includes the encrypted key "C". The Blob 60e includes the encrypted key "A". The Blob 60f includes the encrypted key "B". The Blob 60g includes the encrypted key "C". The Blob 60j includes the encrypted encryption key 62 for the NVRAM4, the encryption key 62 being encrypted using the keys "A" through "C". The Blob 60k includes the encrypted encryption key 62 for the NVRAM4, the encryption key 62 being encrypted using the keys "A" through "C".

[0071] That is, the module configuration in FIG. 17 is different from that in FIG. 12 in that, unlike the Blobs 60d and 60h, the Blobs 60j and 60k have the encrypted encryption key 62 for the NVRAM4 encrypted by using the keys "A" through "C". Because of this configuration, for example, the blob decryption section 23 acquires the keys "A" through "C" and the encrypted encryption key 62 for the NVRAM 4 from the Blobs 60a through 60c and 60j, respectively, and decrypts the acquired encrypted encryption key 62 for the NVRAM 4 using the acquired keys "A" through "C". As a result, the encrypted data 15 in the NVRAM 4 can be decrypted.

[Embodiment 2]

[0072] In the information processing apparatus in above embodiment, should the disk 3 crash, since the encryption key 62 for the NVRAM 4 is to be lost, the encrypted data 15 in the NVRAM 4 can no longer be decrypted. To solve this problem, in an information processing apparatus according to this embodiment 2, a mechanism may be provided that permits decrypting the encrypted data 15 in the NVRAM 4 even when the disk 3 crashes.

[0073] In the information processing apparatus in this embodiment 2, the module configuration of the disk 3 and the information stored in the NVRAM 4 and the HDD 8 are different from those in embodiment 1. FIG. 18 shows an exemplary module configuration of the disk 3 in this embodiment. The disk 3 in FIG. 18 includes the primary modules of the loader 11, the kernel 12, the root file system 13, and the root file system 13; backup modules of the loader 11b, the kernel 12b, the root file system 13b; Blobs 60a through 60d and 601; and the encrypted encryption key 62 for the NVRAM 4. It should be noted that in the information processing apparatus in this embodiment 2, the ROM update for the backup modules of the loader 11b, the kernel 12b, the root file system 13b is not to be performed after the shipment.

[0074] The Blob 60a includes the encrypted key "A". The Blob 60b includes the encrypted key "B". The Blob 60c includes the encrypted key "C". The Blob 60d includes the encrypted key "D". The Blob 601 includes the encrypted encryption key 62 for the NVRAM 4.

[0075] In this configuration, the encryption key 62 for the NVRAM 4 can be decrypted and obtained using the keys "A" through "D" from the Blobs 60a through 60d, respectively, and can be obtained from the Blob 601 corresponding to the boot path of BIOS 10b → loader 11b → kernel 12b → root file system 13b.

[0076] Further, in the information processing apparatus in this embodiment 2, the Blob 601 is stored in the NVRAM 4 and the HDD 8 as shown in FIGS. 19 and 20 so as to respond to the crash of the disk 3. FIGS. 19 and 20 show the configuration of the information stored in the NVRAM 4 and the HDD 8, respectively.

[0077] FIGS. 21 and 22 are drawings showing a process of decrypting the encrypted data 15 in the NVRAM 4. In step S81, primary modules of the loader 11, the kernel 12, the root file system 13, and the backup modules of the loader 11b, the kernel 12b, the root file system 13b are installed in the disk 3.

[0078] In step S82, the ROM update flag control program 22 turns ON a backup flag 71 in the encrypted data 15 in the NVRAM 4. In step S83, the information processing apparatus reboots in a backup mode.

[0079] In step S84, the information processing apparatus boots the loader 11b, the kernel 12b, the root file system 13b (backup mode). In step S85, the blob update program 25 creates a copy of the Blob 601 stored in the NVRAM 4 and stores the created copy in the disk 3. In step S86, the blob update program creates new keys "A" through "D".

[0080] In step S87, the blob update program 25 creates Blobs 80a through 80d including the keys "A" through "D", respectively. In step S88, the blob update program 25 stores the created blobs 80a through 80d in the disk 3. In step S89, the blob decryption section 23 acquires the encryption key 62 from the Blob 601 stored in the NVRAM 4.

[0081] In step S90, the encryption key update program 26 encrypts the encryption key 62 using the keys "A" through "D" and stores the encrypted encryption key 62 in the disk 3. In step S91, the ROM update flag control program 22 turns OFF the backup flag 71 in the encrypted data 15 in the NVRAM 4.

[0082] In the process shown in FIGS. 21 and 22, even when the disk 3 crashes, the encryption key 62 can be acquired from the Blob 601 stored in the NVRAM 4, the Blob 601 corresponding to the boot path of BIOS 10b → loader 11b → kernel 12b → root file system 13b. As a result, the encrypted data 15 in the NVRAM 4 can be decrypted.

[0083] It should be noted that the terms "value storage unit", "encryption information storage units", "information decryption unit", and "encryption information update unit" described in claims herein correspond to the TPM 7, the Blobs 60a through 601, the blob decryption section 23, and the blob update program 25, respectively.

Claims

1. An information processing apparatus including one or more kinds of primary modules (10, 11, 12, 13) necessary to boot the apparatus and one or more kinds of backup modules (10b, 11b, 12b, 13b) to be used when the primary modules (10, 11, 12, 13) fail, so that the information processing apparatus is booted in a manner that when any kind of the primary modules (10, 11, 12, 13) fails, the same kind of backup module (10b, 11b, 12b, 13b) is used, the information processing apparatus comprising, **characterized by**:

   a value storage unit (7) storing values uniquely calculated from the one or more kinds of the primary modules (10, 11, 12, 13) or the backup modules (10b, 11b, 12b, 13b) used when the apparatus is booted;
   an encryption information storage unit (60a, 60b, 60c, 60d) storing information unique to the each kind of the primary or the backup modules, the information being encrypted based on a value calculated from the each kind of the primary modules or the backup modules;
   an information decryption unit (23) decrypting the information unique to the each kind of the primary modules or the backup modules using the values in the value storage unit, the information being stored in the encryption information storage unit; and
   an encryption information update unit (25), when any of the primary modules or the backup modules is updated, encrypting the information unique to the each kind of the primary modules or the backup modules based on a value calculated from the each kind of the primary modules or the backup modules after the update, the information being stored in the encryption information storage unit.

2. The information processing apparatus according to claim 1, wherein
   the information unique to the each kind of the primary modules (10, 11, 12, 13) or the backup modules (10b, 11b, 12b, 13b) is used for encrypting or decrypting an encryption key (A, B, C, D) that is for encrypting or decrypting data used by a user.

3. The information processing apparatus according to claim 2, wherein
   when the encryption key (A, B, C, D) is updated, the updated encryption key is first encrypted using the information unique to the each kind of the primary modules (10, 11, 12, 13) or the backup modules (10b, 11b, 12b, 13b), the information being stored in the encryption information storage unit (60), data used by a user is decrypted using the encryption key that is not updated in the update, and the data used by the user is encrypted using the updated encryption key.

4. The information processing apparatus according to claim 2 or 3, wherein
   the encryption information storage unit (60) stores

the encryption key (A, B, C, D) encrypted using the information unique to the each kind of the primary modules (10, 11, 12, 13) or the backup modules (lOb, 11b, 12b, 13b).

5. The information processing apparatus according to claim 2 or 3, wherein
the encryption information storage unit (60) stores the encryption key (A, B, C, D) encrypted based on a value uniquely calculated from the backup module (10b, 11b, 12b, 13b) that is not updated after the shipment of the apparatus.

6. The information processing apparatus according to claim 5, wherein
when a module storage unit storing the primary modules (10, 11, 12, 13) and the backup modules (lOb, 11b, 12b, 13b) is updated, the apparatus is booted using the backup modules installed in the module storage unit, new information unique to the each kind of the primary modules or the backup modules is created to reform the encryption information storage unit (60), and
the encryption key (A, B, C, D) in the encryption information storage unit is decrypted by the information decryption unit and encrypted using the newly created information unique to the each kind of the primary modules or the backup modules.

7. The information processing apparatus according to claim 1,
wherein the information decryption unit (23) compares a value used for encryption stored in the encryption information storage unit (60) with a value in the value storage unit (7), and
when the value used for encrypting stored in the encryption information storage unit is the same as the value in the value storage unit, the information decryption unit decrypts the information unique to the each kind of the primary (10, 11, 12, 13) or the backup modules (10b, 11b, 12b, 13b), the information being stored in the encryption information storage unit.

8. The information processing apparatus according to any one of claims 1 through 7, wherein
the value storage unit (7) is a Trusted Platform Module (TPM).

9. A software update method for an information processing apparatus including one or more kinds of primary modules necessary to boot the apparatus and one or more kinds of backup modules to be used when the primary modules fail, so that the information processing apparatus is booted in a manner that when any kind of the primary modules fails, the same kind of backup module is used, the software update method being **characterized by**:

a value storing step of storing values in an value storage unit, the values being uniquely calculated from the one or more kinds of the primary modules or the backup modules used when the apparatus is booted;
an encryption information storing step of storing information in an encryption information storage unit, the information being unique to the each kind of the primary or the backup modules, and the information being encrypted based on a value calculated from the each kind of the primary modules or the backup modules;
an information decrypting step of decrypting the information by an information decryption unit, the information being unique to the each kind of the primary modules or the backup modules using the values in the value storage unit, and the information being stored in the encryption information storage unit; and
an encryption information update step of, when any of the primary modules or the backup modules is updated, encrypting information by an encryption information update unit based on a value calculated from the each kind of the updated primary modules or the updated backup modules, the information being unique to the each kind of the primary modules or the backup modules, and the information being stored in the encryption information storage unit.

10. The software update method according to claim 9, wherein
the information unique to the each kind of the primary modules or the backup modules is used for encrypting or decrypting an encryption key that is for encrypting or decrypting data used by a user.

11. The software update method according to claim 10, wherein
when the encryption key is updated, the updated encryption key is first encrypted using the information unique to the each kind of the primary modules or the backup modules, the information being stored in the encryption information storage unit, data used by a user is decrypted using the encryption key that is not updated in the update, and the data used by the user is encrypted using the updated encryption key.

12. The software update method according to claim 10 or 11, wherein
the encryption information storage unit stores the encryption key encrypted using the information unique to the each kind of the primary modules or the backup modules.

13. The software update method according to claim 10 or 11, wherein

the encryption information storage unit stores the encryption key encrypted based on a value uniquely calculated from the backup module that is not updated after the shipment of the apparatus.

14. The software update method according to claim 13, wherein
when a module storage unit storing the primary modules and the backup modules is updated, the apparatus is booted using the backup modules installed in the module storage unit, new information unique to the each kind of the primary modules or the backup modules is created to reform the encryption information storage unit, and
the encryption key in the encryption information storage unit is decrypted by the information decryption unit and encrypted using the newly created information unique to the each kind of the primary modules or the backup modules.

15. The software update method according to claim 9, wherein
the information decryption unit compares a value used for encryption stored in the encryption information storage unit with a value in the value storage unit, and
when the value used for encrypting stored in the encryption information storage unit is the same as the value in the value storage unit, the information decryption unit decrypts the information unique to the each kind of the primary or the backup modules, the information being stored in the encryption information storage unit.

16. The software update method according to any one of claims 9 though 15, wherein
the value storage unit is a Trusted Platform Module (TPM).

17. The information processing apparatus according to claim 1, wherein the information processing apparatus in an image processing apparatus including a plotter section, and scanner section so that the plotter and the scanner sections are booted in a manner that when any kind of the primary modules (10, 11, 12, 13) fails, the same kind of backup module (10b, 11b, 12b, 13b) is used.

Patentansprüche

1. Informationsverarbeitungsvorrichtung, die eine oder mehrere Arten von Primärmodulen (10, 11, 12, 13), die notwendig sind, um die Vorrichtung zu booten, und eine oder mehrere Arten von Sicherungsmodulen (10b, 11b, 12b, 13b), die zu verwenden sind, wenn die Primärmodule (10, 11, 12, 13) ausfallen, enthält, so dass die Informationsverarbeitungsvorrichtung auf eine Art gebootet wird, dass, wenn irgendeine Art der Primärmodule (10, 11, 12, 13) ausfällt, die gleiche Art des Sicherungsmoduls (10b, 11b, 12b, 13b) verwendet wird, wobei die Informationsverarbeitungsvorrichtung **dadurch gekennzeichnet ist, dass** sie umfasst:

eine Wertspeichereinheit (7), die Werte speichert, die eindeutig aus der einen oder den mehreren Arten der Primärmodule (10, 11, 12, 13) oder der Sicherungsmodule (10b, 11b, 12b, 13b), die verwendet werden, wenn die Vorrichtung gebootet wird, berechnet werden;
eine Verschlüsselungsinformations-Speichereinheit (60a, 60b, 60c, 60d), die Informationen speichert, die für jede Art der Primär- oder der Sicherungsmodule eindeutig sind, wobei die Informationen basierend auf einem Wert, der aus jeder Art der Primärmodule oder der Sicherungsmodule berechnet wird, verschlüsselt werden;
eine Informationsentschlüsselungseinheit (23), die die Informationen, die für jede Art der Primärmodule oder der Sicherungsmodule eindeutig sind, unter Verwendung der Werte in der Wertspeichereinheit entschlüsselt, wobei die Informationen in der Verschlüsselungsinformations-Speichereinheit gespeichert sind; und
eine Verschlüsselungsinformations-Aktualisierungseinheit (25) die, wenn irgendeines der Primärmodule oder der Sicherungsmodule aktualisiert wird, die für jede Art der Primärmodule oder der Sicherungsmodule eindeutigen Informationen basierend auf einem Wert verschlüsselt, der aus jeder Art der Primärmodule oder der Sicherungsmodule nach der Aktualisierung berechnet wird, wobei die Informationen in der Verschlüsselungsinformations- Speichereinheit gespeichert sind.

2. Informationsverarbeitungsvorrichtung nach Anspruch 1, wobei
die für jede Art der Primärmodule (10, 11, 12, 13) oder der Sicherungsmodule (10b, 11b, 12b, 13b) eindeutigen Informationen zum Verschlüsseln oder Entschlüsseln eines Verschlüsselungsschlüssels (A, B, C, D) verwendet werden, der zum Verschlüsseln oder Entschlüsseln der durch einen Benutzer verwendeten Daten dient.

3. Informationsverarbeitungsvorrichtung nach Anspruch 2, wobei,
wenn der Verschlüsselungsschlüssel (A, B, C, D) aktualisiert wird, der aktualisierte Verschlüsselungsschlüssel zuerst unter Verwendung der Informationen, die für jede Art der Primärmodule (10, 11, 12, 13) oder der Sicherungsmodule (10b, 11 b, 12b, 13b) eindeutig sind, verschlüsselt wird, wobei die Infor-

mationen in der Verschlüsselungsinformations-Speichereinheit (60) gespeichert sind, die durch einen Benutzer verwendeten Daten unter Verwendung des Verschlüsselungsschlüssels, der bei der Aktualisierung nicht aktualisiert wird, entschlüsselt werden und die durch den Benutzer verwendeten Daten unter Verwendung des aktualisierten Verschlüsselungsschlüssels verschlüsselt werden.

4. Informationsverarbeitungsvorrichtung nach Anspruch 2 oder 3, wobei
die Verschlüsselungsinformations-Speichereinheit (60) den unter Verwendung der Informationen, die für jede Art der Primärmodule (10, 11, 12, 13) oder der Sicherungsmodule (10b, 11b, 12b, 13b) eindeutig sind, verschlüsselten Verschlüsselungsschlüssel (A, B, C, D) speichert.

5. Informationsverarbeitungsvorrichtung nach Anspruch 2 oder 3, wobei
die Verschlüsselungsinformations-Speichereinheit (60) den basierend auf einem aus dem Sicherungsmodul (10b, 11b, 12b, 13b), das nach dem Versand der Vorrichtung nicht aktualisiert wird, eindeutig berechneten Wert verschlüsselten Verschlüsselungsschlüssel (A, B, C, D) speichert.

6. Informationsverarbeitungsvorrichtung nach Anspruch 5, wobei,
wenn eine Modulspeichereinheit, die die Primärmodule (10, 11, 12, 13) und die Sicherungsmodule (10b, 11b, 12b, 13b) speichert, aktualisiert wird, die Vorrichtung unter Verwendung der Sicherungsmodule, die in der Modulspeichereinheit installiert sind, gebootet wird, wobei neue Informationen, die für jede Art der Primärmodule oder der Sicherungsmodule eindeutig sind, erzeugt werden, um die Verschlüsselungsinformations-Speichereinheit (60) zu verbessern, und
der Verschlüsselungsschlüssel (A, B, C, D) in der Verschlüsselungsinformations- Speichereinheit durch die Informationsentschlüsselungseinheit entschlüsselt und unter Verwendung der neu erzeugten Informationen, die für jede Art der Primärmodule oder der Sicherungsmodule eindeutig sind, verschlüsselt wird.

7. Informationsverarbeitungsvorrichtung nach Anspruch 1,
wobei die Informationsentschlüsselungseinheit (23) einen für die Verschlüsselung verwendeten Wert, der in der Verschlüsselungsinformations-Speichereinheit (60) gespeichert ist, mit einem Wert in der Wertspeichereinheit (7) vergleicht, und,
wenn der in der Verschlüsselungsinformations-Speichereinheit gespeicherte Wert, der für die Verschlüsselung verwendet wird, der gleiche Wert wie der Wert in der Wertspeichereinheit ist, die Informationsentschlüsselungseinheit die Informationen, die für jede Art der Primär- (10, 11, 12, 13) oder der Sicherungsmodule (10b, 11b, 12b, 13b) eindeutig sind, entschlüsselt, wobei die Informationen in der Verschlüsselungsinformations-Speichereinheit gespeichert sind.

8. Informationsverarbeitungsvorrichtung nach einem der Ansprüche 1 bis 7, wobei
die Wertspeichereinheit (7) ein Trusted Platform Module (TPM) ist.

9. Softwareaktualisierungsverfahren für eine Informationsverarbeitungsvorrichtung, die eine oder mehrere Arten von Primärmodulen, die notwendig sind, um die Vorrichtung zu booten, und eine oder mehrere Arten von Sicherungsmodulen, die zu verwenden sind, wenn die Primärmodule ausfallen, enthält, so dass die Informationsverarbeitungsvorrichtung auf eine Art gebootet wird, dass, wenn irgendeine Art der Primärmodule ausfällt, die gleiche Art des Sicherungsmoduls verwendet wird, wobei das Softwareaktualisierungsverfahren **gekennzeichnet ist durch**:

einen Wertspeicherschritt des Speicherns von Werten in einer Wertspeichereinheit, wobei die Werte aus der einen oder den mehreren Arten der Primärmodule oder der Sicherungsmodule, die verwendet werden, wenn die Vorrichtung gebootet wird, eindeutig berechnet werden;
einen Verschlüsselungsinformations-Speicherschritt des Speicherns von Informationen in einer Verschlüsselungsinformations-Speichereinheit, wobei die Informationen für jede Art der Primär- oder der Sicherungsmodule eindeutig sind, wobei die Informationen basierend auf einem Wert, der aus jeder Art der Primärmodule oder der Sicherungsmodule berechnet wird, verschlüsselt werden;
einen Informationsentschlüsselungsschritt des Entschlüsselns der Informationen **durch** eine Informationsentschlüsselungseinheit unter Verwendung der Werte in der Wertspeichereinheit, wobei die Informationen für jede Art der Primärmodule oder der Sicherungsmodule eindeutig sind, wobei die Informationen in der Verschlüsselungsinformations-Speichereinheit gespeichert sind; und
einen Verschlüsselungsinformations-Aktualisierungsschritt des Verschlüsselns der Informationen **durch** eine Verschlüsselungsinformations-Aktualisierungseinheit, wenn irgendeines der Primärmodule oder der Sicherungsmodule aktualisiert wird, basierend auf einem Wert, der aus jeder Art der aktualisierten Primärmodule oder der aktualisierten Sicherungsmodule berechnet wird, wobei die Informationen für jede

Art der Primärmodule oder der Sicherungsmodule eindeutig sind, wobei die Informationen in der Verschlüsselungsinformations-Speichereinheit gespeichert sind.

10. Softwareaktualisierungsverfahren nach Anspruch 9, wobei
die für jede Art der Primärmodule oder der Sicherungsmodule eindeutigen Informationen zum Verschlüsseln oder Entschlüsseln eines Verschlüsselungsschlüssels verwendet werden, der zum Verschlüsseln oder Entschlüsseln der durch einen Benutzer verwendeten Daten dient.

11. Softwareaktualisierungsverfahren nach Anspruch 10, wobei
wenn der Verschlüsselungsschlüssel aktualisiert wird, der aktualisierte Verschlüsselungsschlüssel zuerst unter Verwendung der Informationen, die für jede Art der Primärmodule oder der Sicherungsmodule eindeutig sind, verschlüsselt wird, wobei die Informationen in der Verschlüsselungsinformations-Speichereinheit gespeichert sind, die durch einen Benutzer verwendeten Daten unter Verwendung des Verschlüsselungsschlüssels, der bei der Aktualisierung nicht aktualisiert wird, entschlüsselt werden und die durch den Benutzer verwendeten Daten unter Verwendung des aktualisierten Verschlüsselungsschlüssels verschlüsselt werden.

12. Softwareaktualisierungsverfahren nach Anspruch 10 oder 11, wobei
die Verschlüsselungsinformations-Speichereinheit den unter Verwendung der Informationen, die für jede Art der Primärmodule oder der Sicherungsmodule eindeutig sind, verschlüsselten Verschlüsselungsschlüssel speichert.

13. Softwareaktualisierungsverfahren nach Anspruch 10 oder 11, wobei
die Verschlüsselungsinformations-Speichereinheit den basierend auf einem aus dem Sicherungsmodul, das nach dem Versand der Vorrichtung nicht aktualisiert wird, eindeutig berechneten Wert verschlüsselten Verschlüsselungsschlüssel speichert.

14. Softwareaktualisierungsverfahren nach Anspruch 13, wobei
wenn eine Modulspeichereinheit, die die Primärmodule und die Sicherungsmodule speichert, aktualisiert wird, die Vorrichtung unter Verwendung der Sicherungsmodule, die in der Modulspeichereinheit installiert sind, gebootet wird, wobei neue Informationen, die für jede Art der Primärmodule oder der Sicherungsmodule eindeutig sind, erzeugt werden, um die Verschlüsselungsinformations-Speichereinheit zu verbessern, und
der Verschlüsselungsschlüssel in der Verschlüsselungsinformations-Speichereinheit durch die Informationsentschlüsselungseinheit entschlüsselt und unter Verwendung der neu erzeugten Informationen, die für jede Art der Primärmodule oder der Sicherungsmodule eindeutig sind, verschlüsselt wird.

15. Softwareaktualisierungsverfahren nach Anspruch 9, wobei
die Informationsentschlüsselungseinheit einen für die Verschlüsselung verwendeten Wert, der in der Verschlüsselungsinformations-Speichereinheit gespeichert ist, mit einem Wert in der Wertspeichereinheit vergleicht, und,
wenn der in der Verschlüsselungsinformations-Speichereinheit gespeicherte Wert, der für die Verschlüsselung verwendet wird, der gleiche Wert wie der Wert in der Wertspeichereinheit ist, die Informationsentschlüsselungseinheit die Informationen, die für jede Art der Primär- oder der Sicherungsmodule eindeutig sind, entschlüsselt, wobei die Informationen in der Verschlüsselungsinformations-Speichereinheit gespeichert sind.

16. Softwareaktualisierungsverfahren nach einem der Ansprüche 9 bis 15, wobei die Wertspeichereinheit ein Trusted Platform Module (TPM) ist.

17. Informationsverarbeitungsvorrichtung nach Anspruch 1, wobei die Informationsverarbeitungsvorrichtung in einer Bildverarbeitungsvorrichtung einen Plotter-Abschnitt und einen Scanner-Abschnitt enthält, so dass der Plotter- und der Scanner-Abschnitt auf eine Art gebootet werden, dass, wenn irgendeine Art der Primärmodule (10, 11, 12, 13) ausfällt, die gleiche Art des Sicherungsmoduls (10b, 11 b, 12b, 13b) verwendet wird.

Revendications

1. Appareil de traitement d'information comprenant un ou plusieurs types de modules primaires (10, 11, 12, 13) nécessaires pour initialiser l'appareil et un ou plusieurs types de modules de sauvegarde (10b, 11b, 12b, 13b) à utiliser lorsque les modules primaires (10, 11, 12, 13) sont en panne, de sorte que l'appareil de traitement d'information est initialisé de manière que, lorsque l'un quelconque des modules primaires (10, 11, 12, 13) tombe en panne, le même type de modules de sauvegarde (10b, 11b, 12b, 13b) est utilisé, l'appareil de traitement d'information étant **caractérisé par** :

une unité de stockage de valeurs (7) mémorisant des valeurs uniquement calculées à partir de l'un ou de plusieurs types de modules primaires (10, 11, 12, 13) ou de modules de sauvegarde (10b, 11b, 12b, 13b) utilisés lorsque l'ap-

pareil est initialisé ;
une unité de stockage d'information de cryptage (60a, 60b, 60c, 60d) mémorisant une information unique pour chaque type de modules primaires ou de modules de sauvegarde, l'information étant cryptées en fonction d'une valeur calculée à partir de chaque type de modules primaires ou de modules de sauvegarde ;
une unité de décryptage d'information (23) décryptant l'information unique pour chaque type de modules primaires ou de modules de sauvegarde en utilisant les valeurs dans l'unité de stockage de valeurs, l'information étant mémorisée dans l'unité de stockage d'information de cryptage ; et
une unité de mise à jour d'information de cryptage (25), lorsque l'un quelconque des modules primaires ou des modules de sauvegarde est mis à jour, cryptant l'information unique pour chaque type de modules primaires ou de modules de sauvegarde en fonction d'une valeur calculée à partir de chaque type de modules primaires ou de modules de sauvegarde après la mise à jour, l'information étant mémorisée dans l'unité de stockage d'information de cryptage.

2. Appareil de traitement d'information selon la revendication 1, dans lequel
l'information unique pour chaque type de modules primaires (10, 11, 12, 13) ou de modules de sauvegarde (10b, 11b, 12b, 13b) est utilisée pour crypter ou décrypter une clé de cryptage (A, B, C, D) qui est conçue pour crypter ou décrypter des données utilisées par un utilisateur.

3. Appareil de traitement d'information selon la revendication 2, dans lequel
lorsque la clé de cryptage (A, B, C, D) est mise à jour, la clé de cryptage mise à jour est tout d'abord cryptée en utilisant l'information unique pour chaque type de modules primaires (10, 11, 12, 13) ou de modules de sauvegarde (10b, 11b, 12b, 13b), l'information étant mémorisée dans l'unité de stockage d'information de cryptage (60), des données utilisées par un utilisateur sont décryptées en utilisant la clé de cryptage qui n'est pas mise à jour lors de la mise à jour, et les données utilisées par l'utilisateur sont cryptées en utilisant la clé de cryptage mise à jour.

4. Appareil de traitement d'information selon la revendication 2 ou 3, dans lequel
l'unité de stockage d'information de cryptage (60) mémorise la clé de cryptage (A, B, C, D) cryptée en utilisant l'information unique pour chaque type de modules primaires (10, 11, 12, 13) ou de modules de sauvegarde (10b, 11b, 12b, 13b).

5. Appareil de traitement d'information selon la revendication 2 ou 3, dans lequel
l'unité de stockage d'information de cryptage (60) mémorise la clé de cryptage (A, B, C, D) cryptée en fonction d'une valeur uniquement calculée à partir du module de sauvegarde (10b, 11b, 12b, 13b) qui n'est pas mis à jour après la livraison de l'appareil.

6. Appareil de traitement d'information selon la revendication 5, dans lequel
lorsqu'une unité de stockage de modules mémorisant les modules primaires (10, 11, 12, 13) et les modules de sauvegarde (10b, 11b, 12b, 13b) est mise à jour, l'appareil est initialisé en utilisant les modules de sauvegarde installés dans l'unité de stockage de modules, une nouvelle information unique pour chaque type de modules primaires ou de modules de sauvegarde est créée pour reformer l'unité de stockage d'information de cryptage (60), et
la clé de cryptage (A, B, C, D) dans l'unité de stockage d'information de cryptage est décryptée par l'unité de décryptage d'information et cryptée en utilisant l'information nouvellement créée unique pour chaque type de modules primaires ou de modules de sauvegarde.

7. Appareil de traitement d'information selon la revendication 1,
dans lequel l'unité de décryptage d'information (23) compare une valeur utilisée pour le cryptage mémorisée dans l'unité de stockage d'information de cryptage (60) à une valeur dans l'unité de stockage de valeur (7), et
lorsque la valeur utilisée pour le cryptage mémorisée dans l'unité de stockage d'information de cryptage est la même que la valeur dans l'unité de stockage de valeur, l'unité de décryptage d'information décrypte l'information unique pour chaque type de modules primaires (10, 11, 12, 13) ou de modules de sauvegarde (10b, 11b, 12b, 13b), l'information étant mémorisée dans l'unité de stockage d'information de cryptage.

8. Appareil de traitement d'information selon l'une quelconque des revendications 1 à 7, dans lequel
l'unité de stockage de valeur (7) est un Module de Plate-forme Sécurisée (TPM).

9. Procédé de mise à jour de logiciel pour un appareil de traitement d'information comprenant un ou plusieurs types de modules primaires nécessaires pour initialiser appareil et un ou plusieurs types de modules de sauvegarde utilisés lorsque les modules primaires tombent en panne, de sorte que l'appareil de traitement d'information est initialisé de manière que, lorsqu'un type quelconque de modules primaires tombe en panne, le même type de modules de sauvegarde est utilisé, le procédé de mise à jour de

logiciel étant **caractérisé par** :

une étape de mémorisation de valeur pour mémoriser des valeurs dans une unité de stockage de valeur, les valeurs étant uniquement calculées à partir d'un ou de plusieurs types de modules primaires ou de modules de sauvegarde utilisés lorsque l'appareil est initialisé ;
une étape de stockage d'information de cryptage pour mémoriser une information dans une unité de stockage d'information de cryptage, l'information étant unique pour chaque type de modules primaires ou de modules de sauvegarde, et l'information étant cryptée en fonction d'une valeur calculée à partir de chaque type de modules primaires ou de modules de sauvegarde ;
une étape de décryptage d'information pour décrypter l'information par une unité de décryptage d'information, l'information étant unique pour chaque type de modules primaires ou de modules de sauvegarde en utilisant les valeurs dans l'unité de stockage de valeurs, et l'information étant mémorisée dans l'unité de stockage d'information de cryptage ; et
une étape de mise à jour d'information de cryptage pour, lorsque l'un quelconque des modules primaires ou des modules de sauvegarde est mis à jour, crypter une information par une unité de mise à jour d'information de cryptage en fonction d'une valeur calculée à partir de chaque type de modules primaires mis à jour ou de modules de sauvegarde mis à jour, l'information étant unique pour chaque type de modules primaires ou de modules de sauvegarde, et l'information étant mémorisée dans l'unité de stockage d'information de cryptage.

10. Procédé de mise à jour de logiciel selon la revendication 9, dans lequel
l'information unique pour chaque type de modules primaires ou de modules de sauvegarde est utilisée pour crypter ou décrypter une clé de cryptage qui est conçue pour crypter ou décrypter des données utilisées par un utilisateur.

11. Procédé de mise à jour de logiciel selon la revendication 10, dans lequel
lorsque la clé de cryptage est mise à jour, la clé de cryptage mise à jour est tout d'abord cryptée en utilisant une information unique pour chaque type de modules primaires ou de modules de sauvegarde, l'information étant mémorisée dans l'unité de stockage d'information de cryptage, les données utilisées par un utilisateur sont décryptées en utilisant la clé de cryptage qui n'est pas mise à jour lors de la mise à jour, et les données utilisées par l'utilisateur sont cryptées en utilisant la clé de cryptage mise à jour.

12. Procédé de mise à jour de logiciel selon la revendication 10 ou 11, dans lequel
l'unité de stockage d'information de cryptage mémorise la clé de cryptage cryptée en utilisant l'information unique pour chaque type de modules primaires ou de modules de sauvegarde.

13. Procédé de mise à jour de logiciel selon la revendication 10 ou 11, dans lequel
l'unité de stockage d'information de cryptage mémorise la clé de cryptage cryptée en fonction d'une valeur uniquement calculée à partir du module de sauvegarde qui n'est pas mis à jour après la livraison de l'appareil.

14. Procédé de mise à jour de logiciel selon la revendication 13, dans lequel
lorsqu'une unité de stockage de modules mémorisant les modules primaires et les modules de sauvegarde est mise à jour, l'appareil est initialisé en utilisant les modules de sauvegarde installés dans l'unité de stockage de modules, une nouvelle information unique pour chaque type de modules primaires ou de module de sauvegarde est créée pour reformer l'unité de stockage d'information de cryptage, et
la clé de cryptage dans l'unité de stockage d'information de cryptage est décryptée par l'unité de décryptage d'information décryptée en utilisant l'information nouvellement créée unique pour chaque type de modules primaires ou de modules de sauvegarde.

15. Procédé de mise à jour de logiciel selon la revendication 9, dans lequel
l'unité de décryptage d'information compare une valeur utilisée pour le cryptage mémorisée dans l'unité de stockage d'information de cryptage à une valeur dans l'unité de stockage de valeurs, et
lorsque la valeur utilisée pour le cryptage mémorisée dans l'unité de stockage d'information de cryptage est la même que la valeur dans l'unité de stockage de valeurs, l'unité de décryptage d'information décrypte l'information unique pour chaque type de modules primaires ou de modules de sauvegarde, l'information étant mémorisée dans l'unité de stockage d'information de cryptage.

16. Procédé de mise à jour de logiciel selon l'une quelconque des revendications 9 à 15, dans lequel
l'unité de stockage de valeur est un Module de Plateforme Sécurisée (TPM).

17. Appareil de traitement d'information selon la revendication 1, dans lequel l'appareil de traitement d'information est un appareil de traitement d'image comprenant une section de traçage, et une section de scanner de sorte que les sections de traçage et de

scanner sont initialisées de manière que, lorsque l'un quelconque des modules primaires (10, 11, 12, 13) tombe en panne, le même type de modules de sauvegarde (10b, 11b, 12b, 13b) est utilisé.

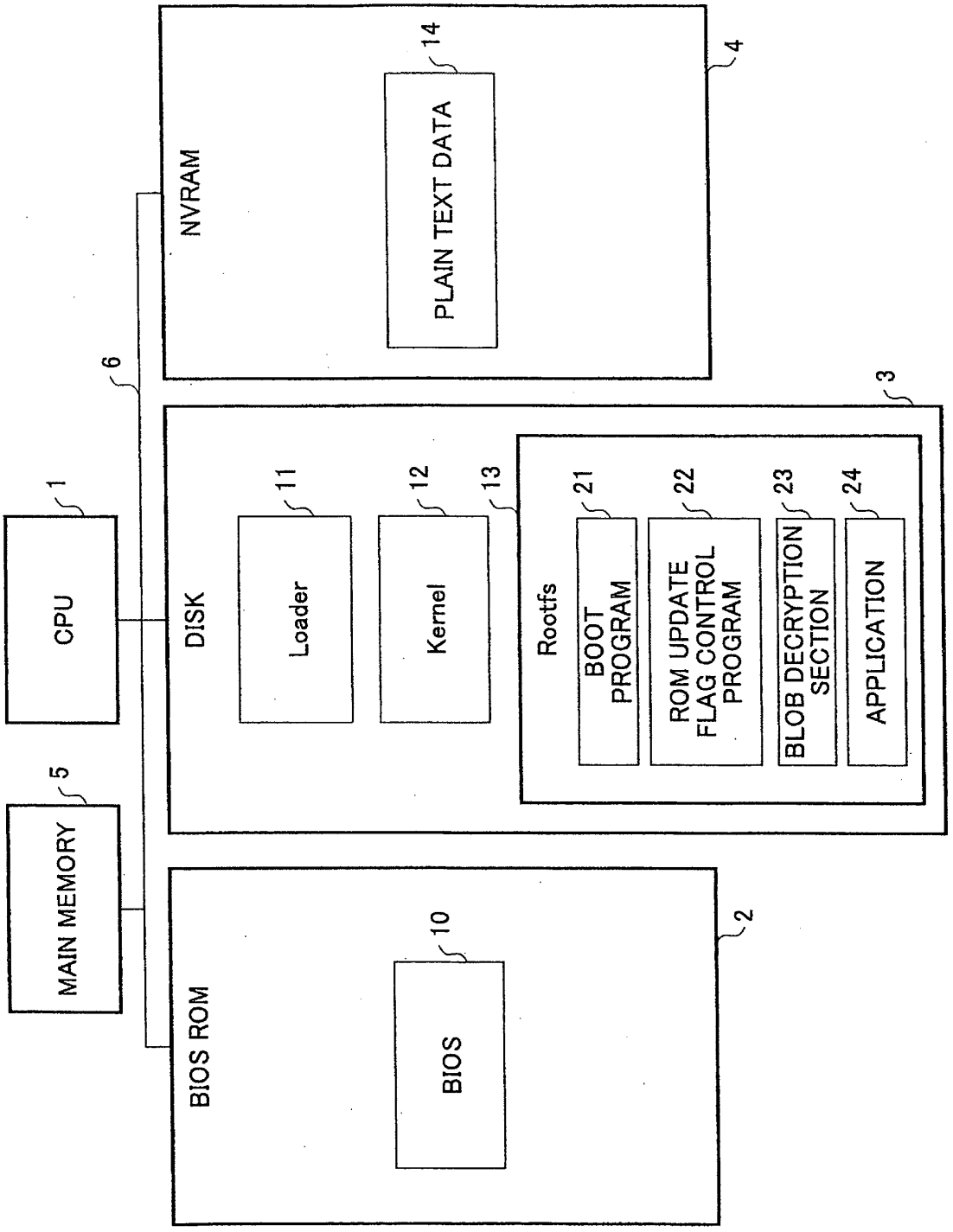

CPU
1
MAIN MEMORY
5
6
BIOS ROM
BIOS
10
2
NVRAM
PLAIN TEXT DATA
14
4
DISK
3
Loader
11
Kernel
12
13
Rootfs
BOOT PROGRAM
21
ROM UPDATE FLAG CONTROL PROGRAM
22
BLOB DECRYPTION SECTION
23
APPLICATION
24

FIG.1

FIG.2

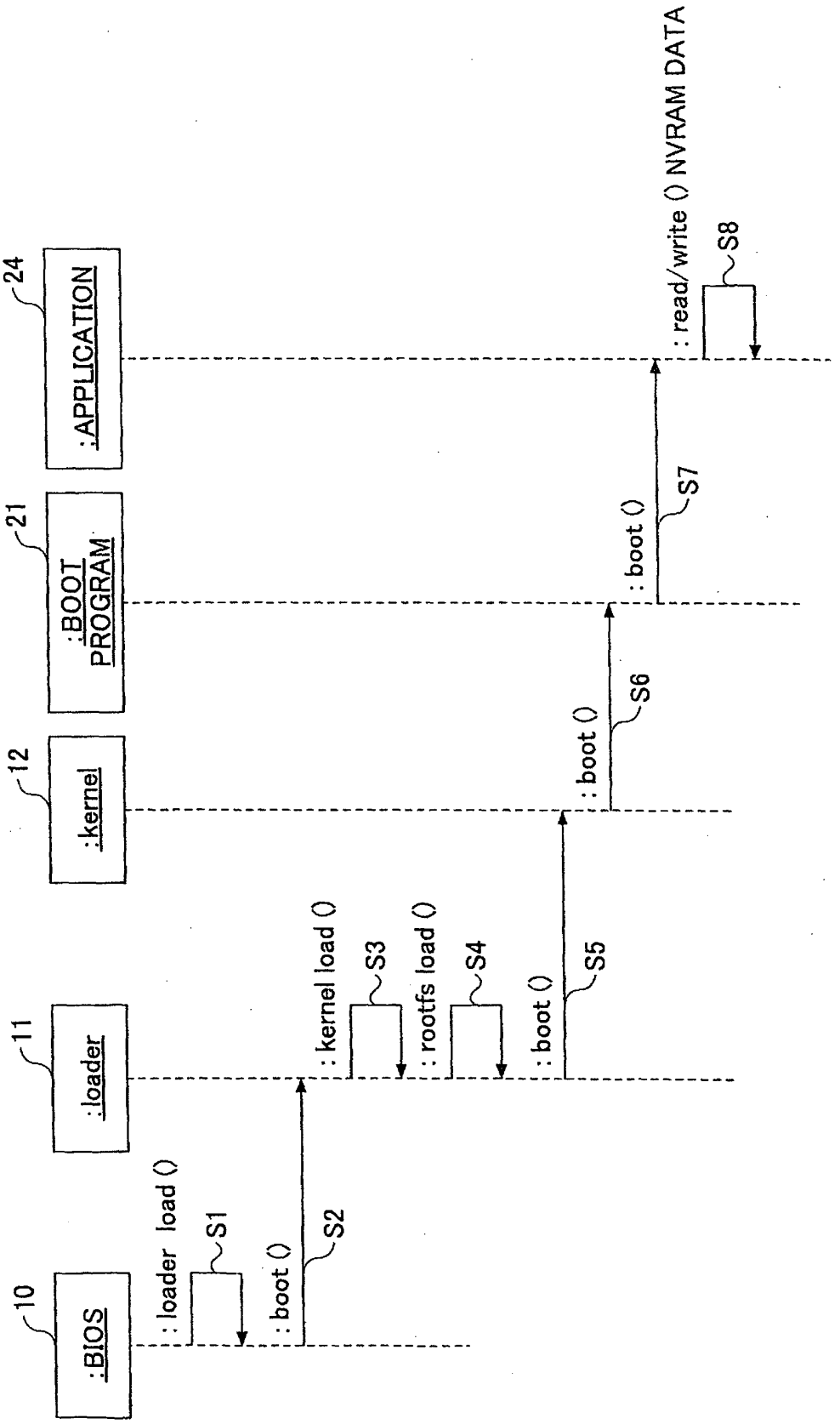

10
:BIOS
11
:loader
12
:kernel
21
:BOOT PROGRAM
24
:APPLICATION
: loader load ()
S1
: boot ()
S2
: kernel load ()
S3
: rootfs load ()
S4
: boot ()
S5
: boot ()
S6
: boot ()
S7
: read/write () NVRAM DATA
S8

FIG.3

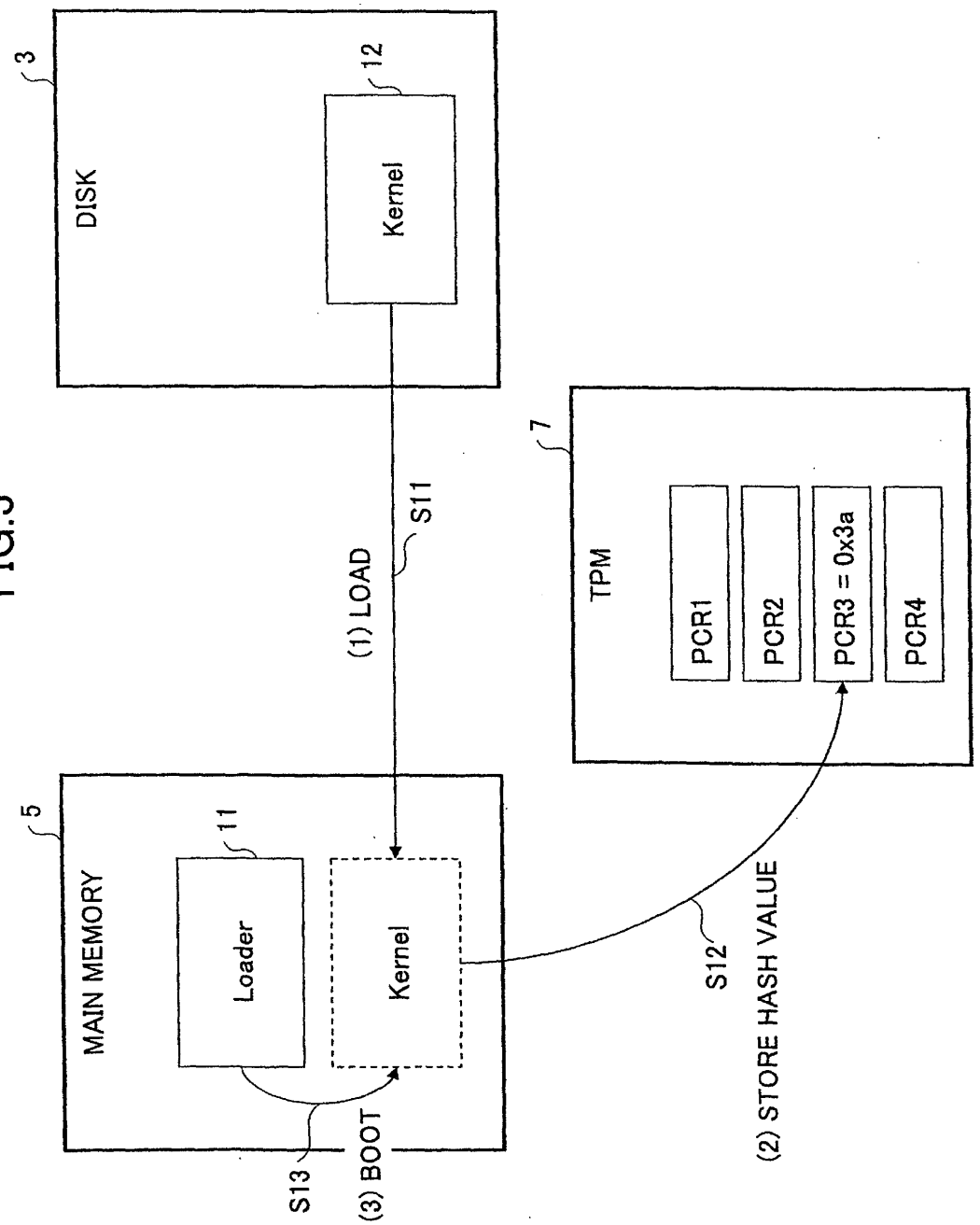

3
DISK
12
Kernel
(1) LOAD
S11
7
TPM
PCR1
PCR2
PCR3 = 0x3a
PCR4
5
MAIN MEMORY
11
Loader
Kernel
S12
(2) STORE HASH VALUE
S13
(3) BOOT

FIG.4

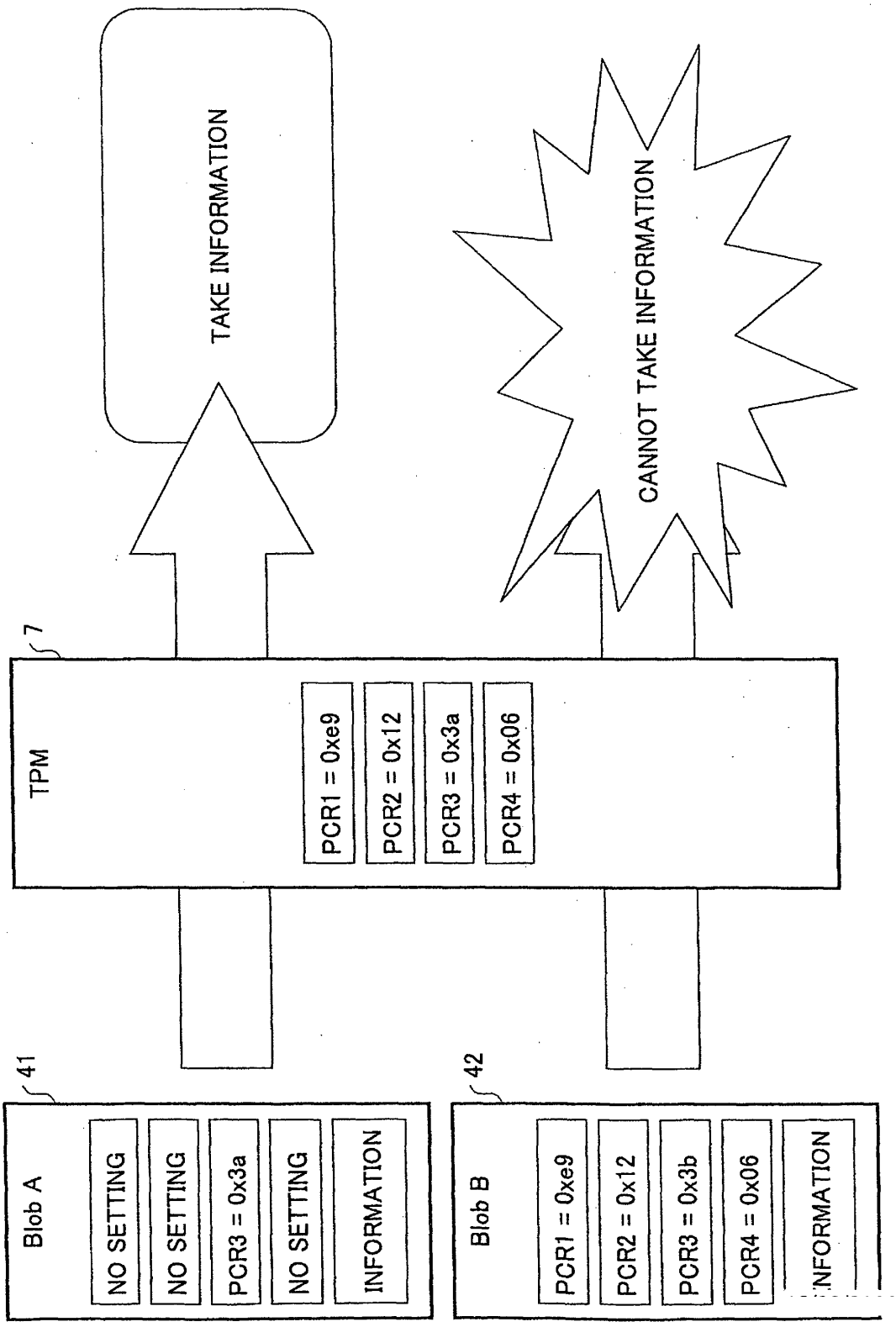

TAKE INFORMATION
CANNOT TAKE INFORMATION
7
TPM
PCR1 = 0xe9
PCR2 = 0x12
PCR3 = 0x3a
PCR4 = 0x06
41
Blob A
NO SETTING
NO SETTING
PCR3 = 0x3a
NO SETTING
INFORMATION
42
Blob B
PCR1 = 0xe9
PCR2 = 0x12
PCR3 = 0x3b
PCR4 = 0x06
INFORMATION

FIG.5

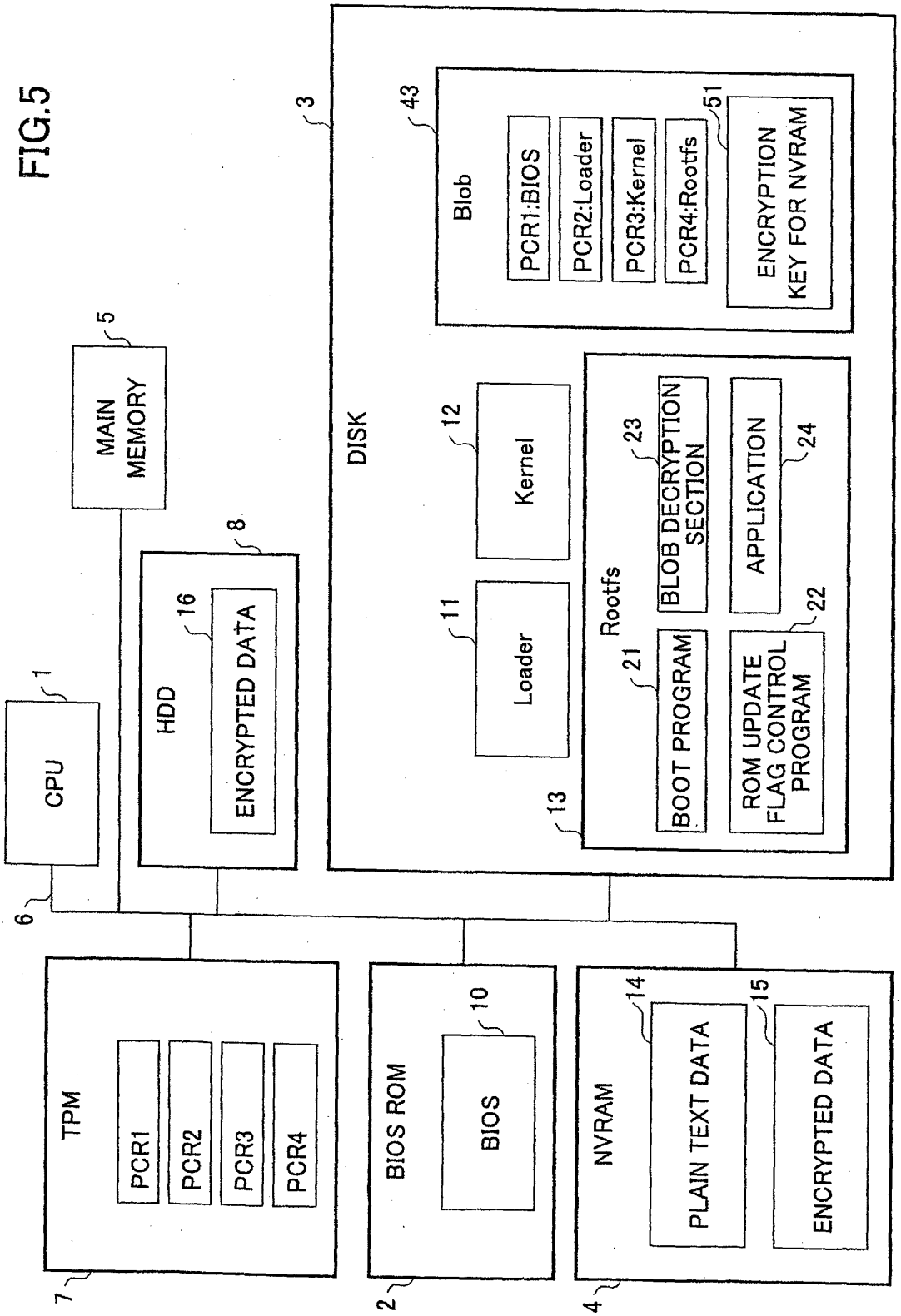

CPU
1
MAIN MEMORY
5
HDD
8
ENCRYPTED DATA
16
6
DISK
3
Blob
43
PCR1:BIOS
PCR2:Loader
PCR3:Kernel
PCR4:Rootfs
51
ENCRYPTION KEY FOR NVRAM
Kernel
12
Loader
11
Rootfs
13
BOOT PROGRAM
21
BLOB DECRYPTION SECTION
23
APPLICATION
24
ROM UPDATE FLAG CONTROL PROGRAM
22
TPM
PCR1
PCR2
PCR3
PCR4
7
BIOS ROM
BIOS
10
2
NVRAM
PLAIN TEXT DATA
14
ENCRYPTED DATA
15
4

FIG.6

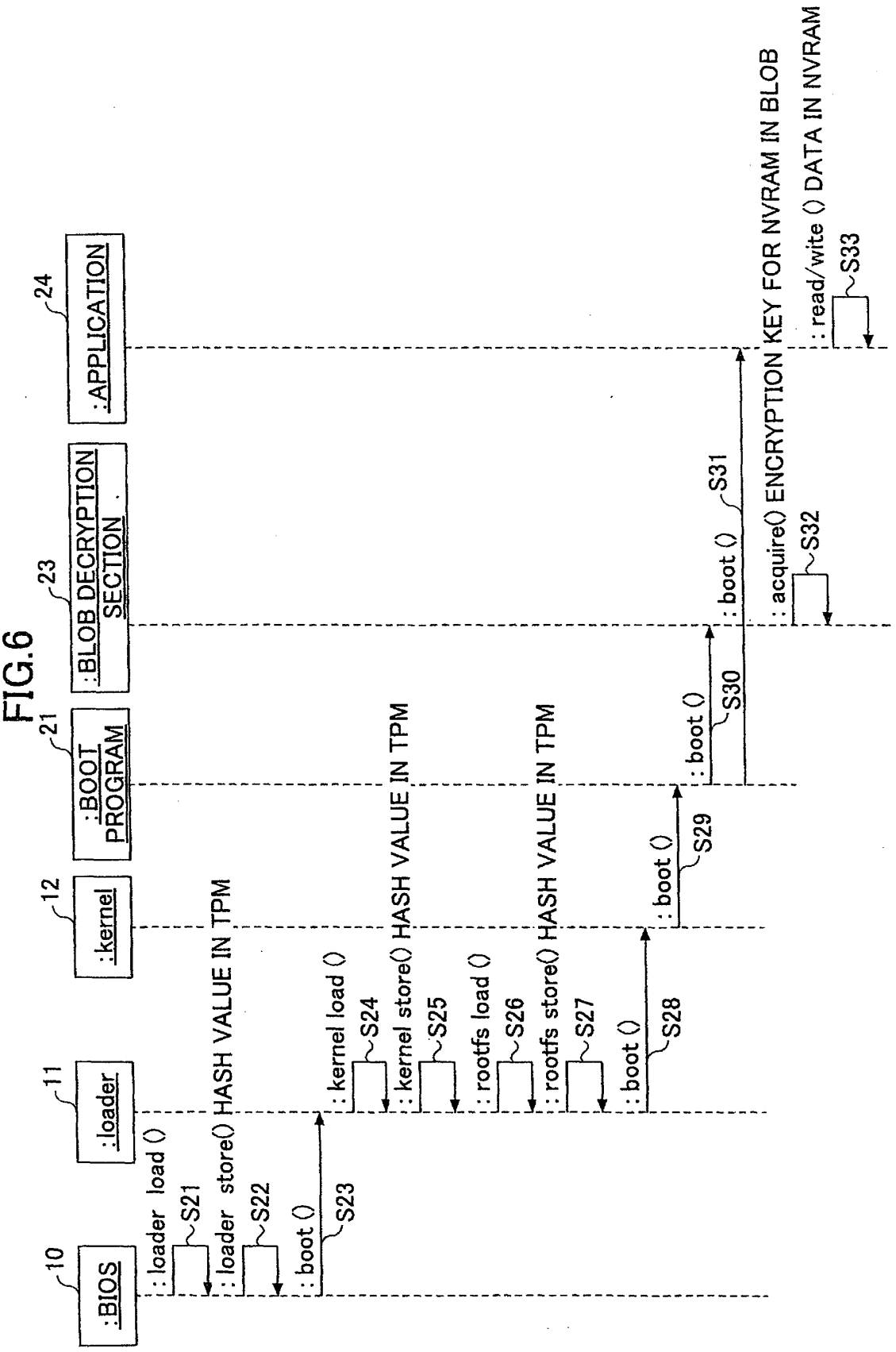

10
:BIOS
11
:loader
12
:kernel
21
:BOOT PROGRAM
23
:BLOB DECRYPTION SECTION
24
:APPLICATION
: loader load ()
S21
: loader store()
HASH VALUE IN TPM
S22
: boot ()
S23
: kernel load ()
S24
: kernel store()
HASH VALUE IN TPM
S25
: rootfs load ()
S26
: rootfs store()
HASH VALUE IN TPM
S27
: boot ()
S28
: boot ()
S29
: boot ()
S30
: boot ()
S31
: acquire() ENCRYPTION KEY FOR NVRAM IN BLOB
S32
: read/wite () DATA IN NVRAM
S33

FIG.7

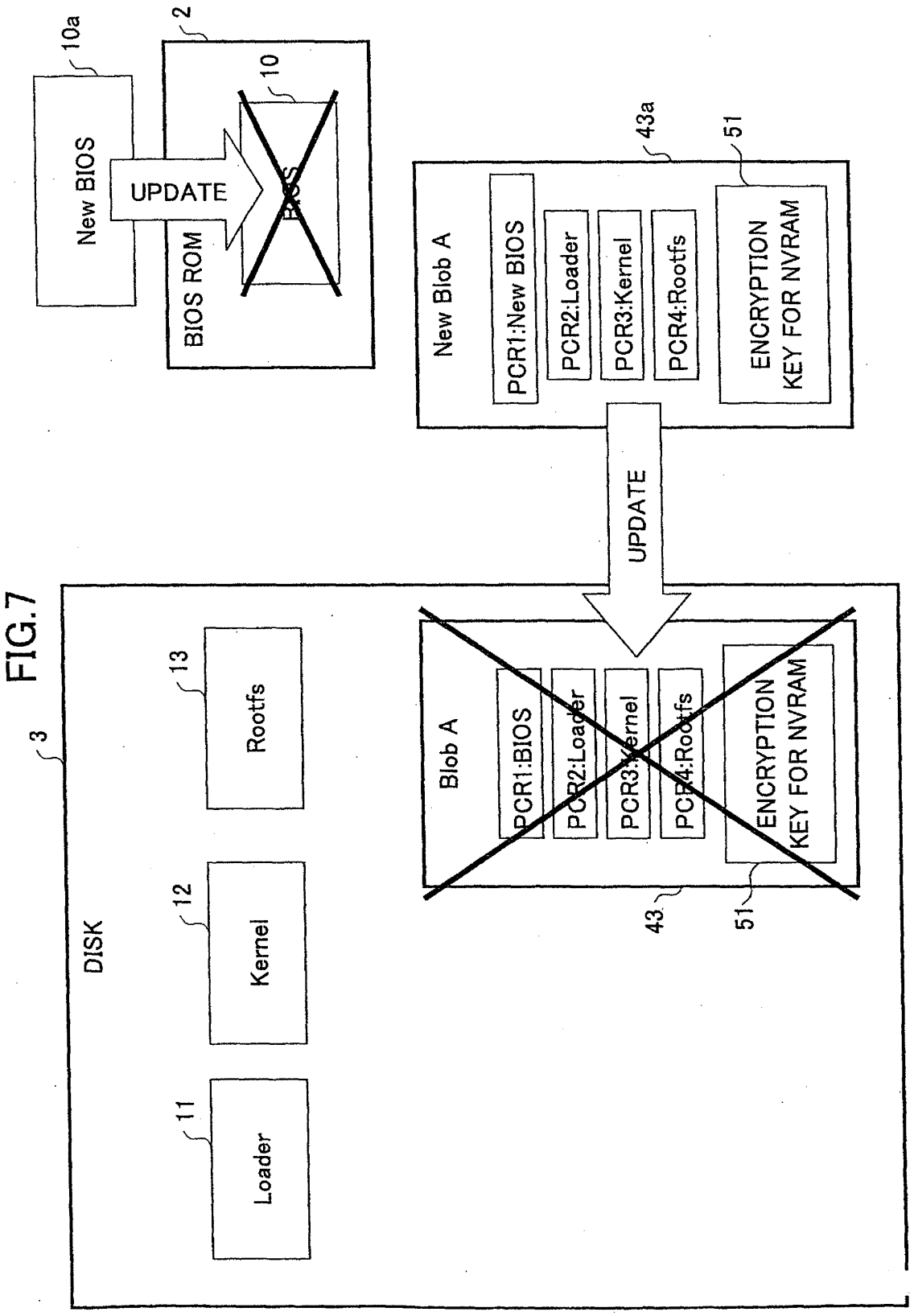

10a
New BIOS
2
10
UPDATE
BIOS ROM
New Blob A
43a
51
PCR1:New BIOS
PCR2:Loader
PCR3:Kernel
PCR4:Rootfs
ENCRYPTION KEY FOR NVRAM
UPDATE
3
DISK
13
Rootfs
12
Kernel
11
Loader
Blob A
PCR1:BIOS
PCR2:Loader
PCR3:Kernel
PCR4:Rootfs
ENCRYPTION KEY FOR NVRAM
43
51

FIG.8

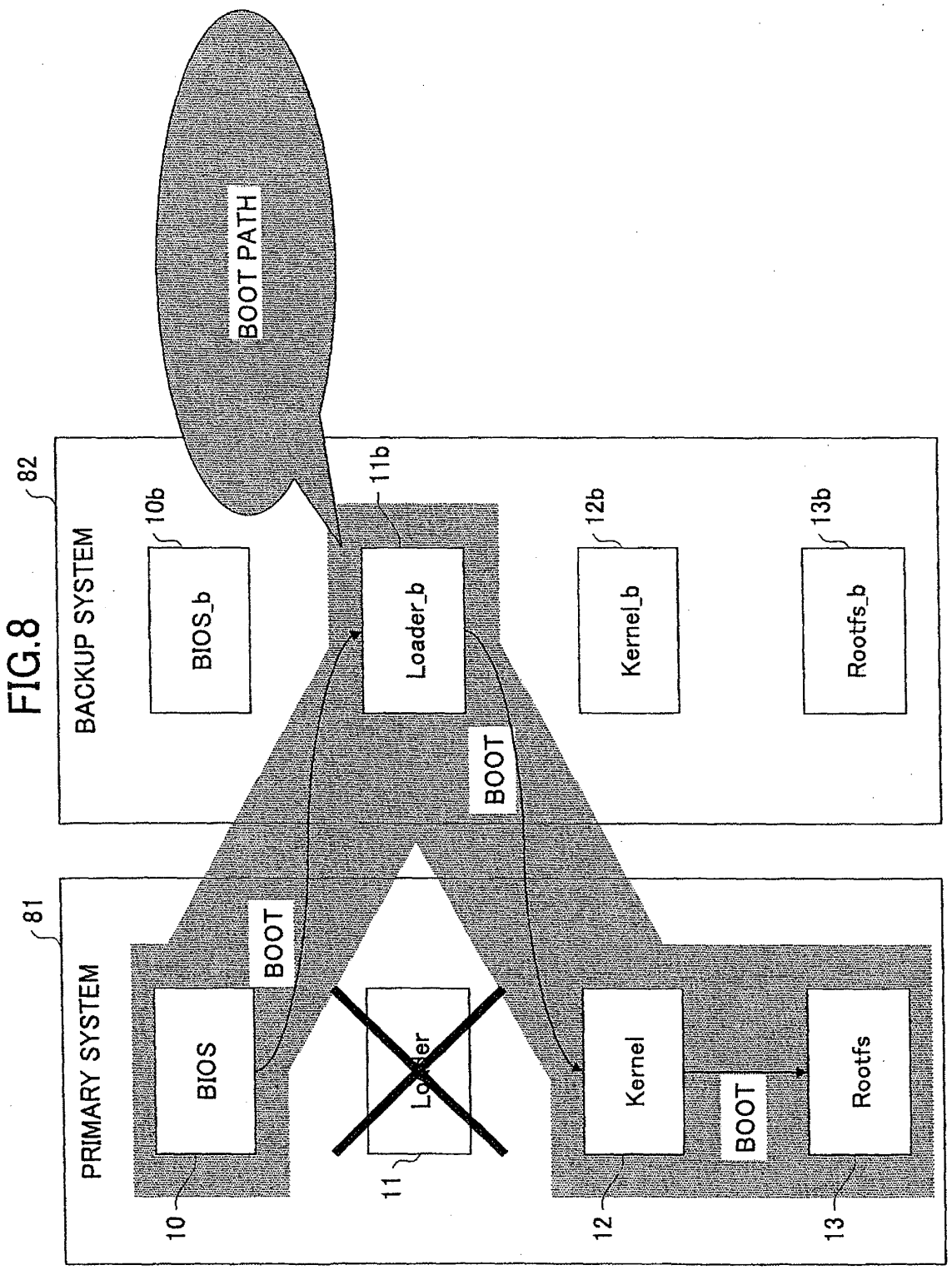

BOOT PATH
82
BACKUP SYSTEM
10b
BIOS_b
11b
Loader_b
12b
Kernel_b
13b
Rootfs_b
BOOT
81
PRIMARY SYSTEM
BIOS
BOOT
Loader
Kernel
BOOT
Rootfs
10
11
12
13

FIG.9

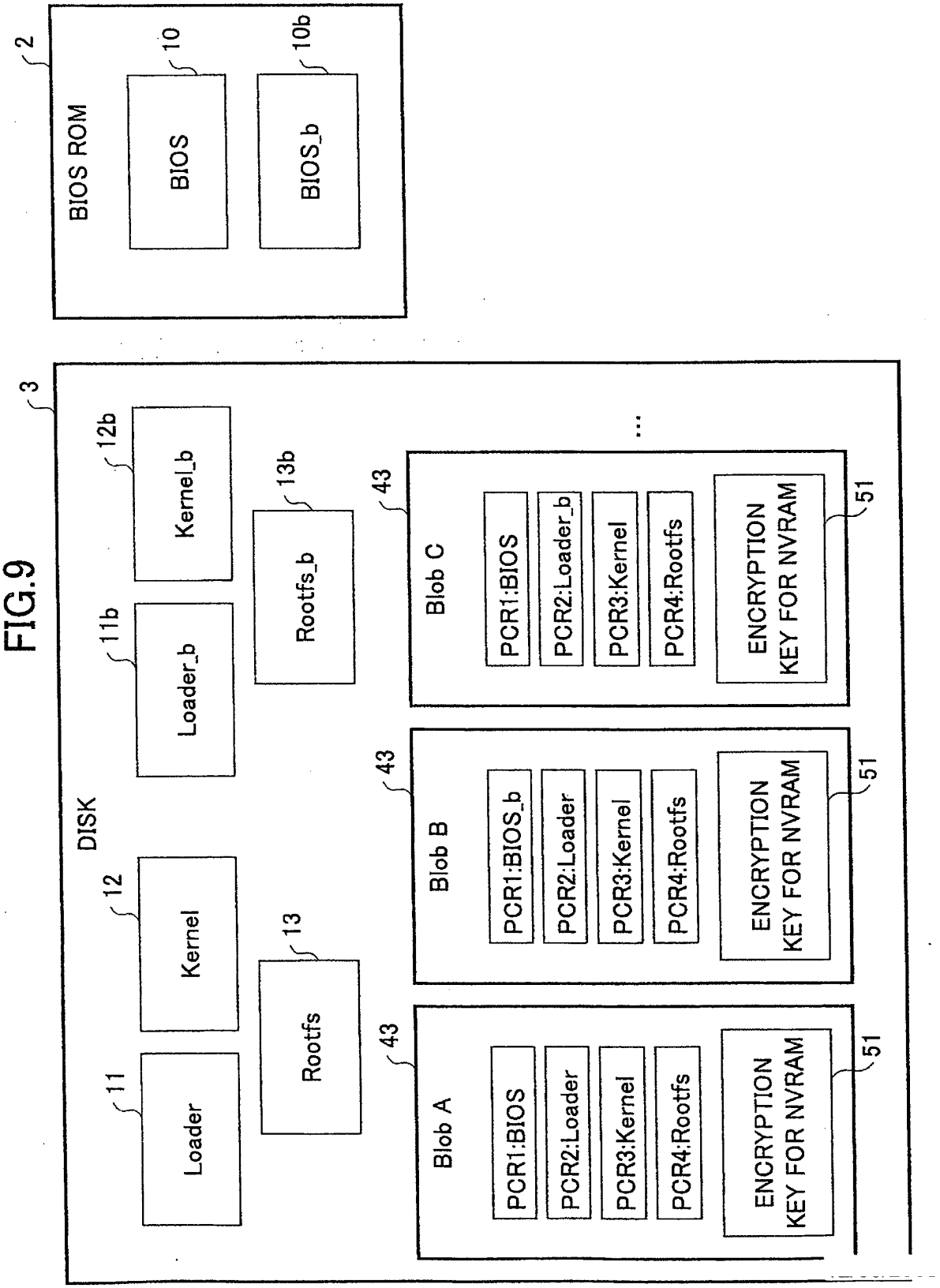

2
BIOS ROM
10
BIOS
10b
BIOS_b
3
DISK
11
Loader
12
Kernel
13
Rootfs
11b
Loader_b
12b
Kernel_b
13b
Rootfs_b
43
Blob A
PCR1:BIOS
PCR2:Loader
PCR3:Kernel
PCR4:Rootfs
ENCRYPTION KEY FOR NVRAM
51
43
Blob B
PCR1:BIOS_b
PCR2:Loader
PCR3:Kernel
PCR4:Rootfs
ENCRYPTION KEY FOR NVRAM
51
43
Blob C
PCR1:BIOS
PCR2:Loader_b
PCR3:Kernel
PCR4:Rootfs
ENCRYPTION KEY FOR NVRAM
51
...

FIG.10

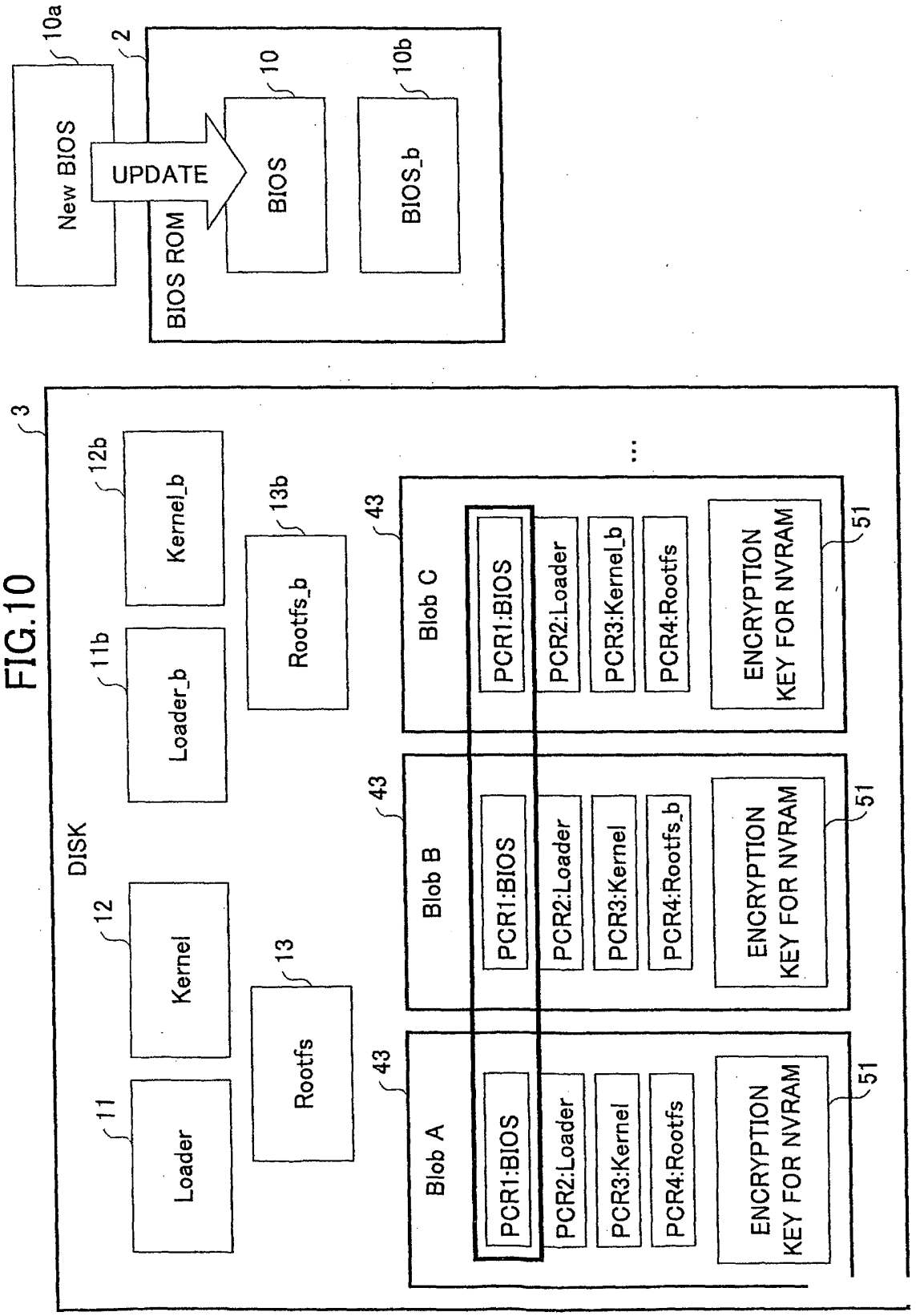

10a
2
New BIOS
UPDATE
BIOS ROM
10
BIOS
10b
BIOS_b
3
DISK
11
Loader
12
Kernel
13
Rootfs
11b
Loader_b
12b
Kernel_b
13b
Rootfs_b
43
Blob A
PCR1:BIOS
PCR2:Loader
PCR3:Kernel
PCR4:Rootfs
ENCRYPTION KEY FOR NVRAM
51
Blob B
PCR1:BIOS
PCR2:Loader
PCR3:Kernel
PCR4:Rootfs_b
ENCRYPTION KEY FOR NVRAM
51
Blob C
PCR1:BIOS
PCR2:Loader
PCR3:Kernel_b
PCR4:Rootfs
ENCRYPTION KEY FOR NVRAM
51
...

FIG.11

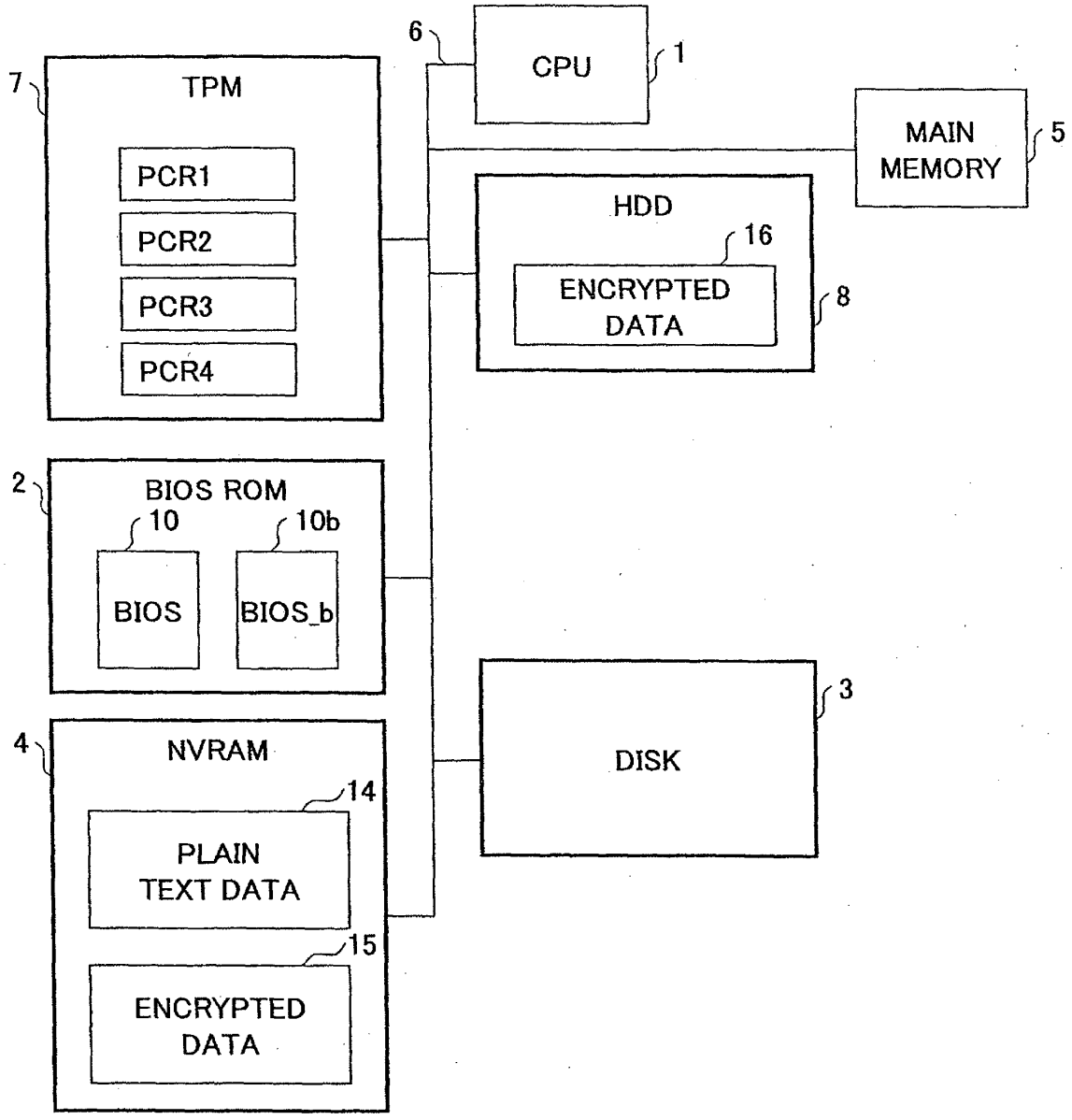

6
CPU
1
7
TPM
PCR1
PCR2
PCR3
PCR4
MAIN MEMORY
5
HDD
16
ENCRYPTED DATA
8
2
BIOS ROM
10
10b
BIOS
BIOS_b
3
DISK
4
NVRAM
14
PLAIN TEXT DATA
15
ENCRYPTED DATA

FIG.12

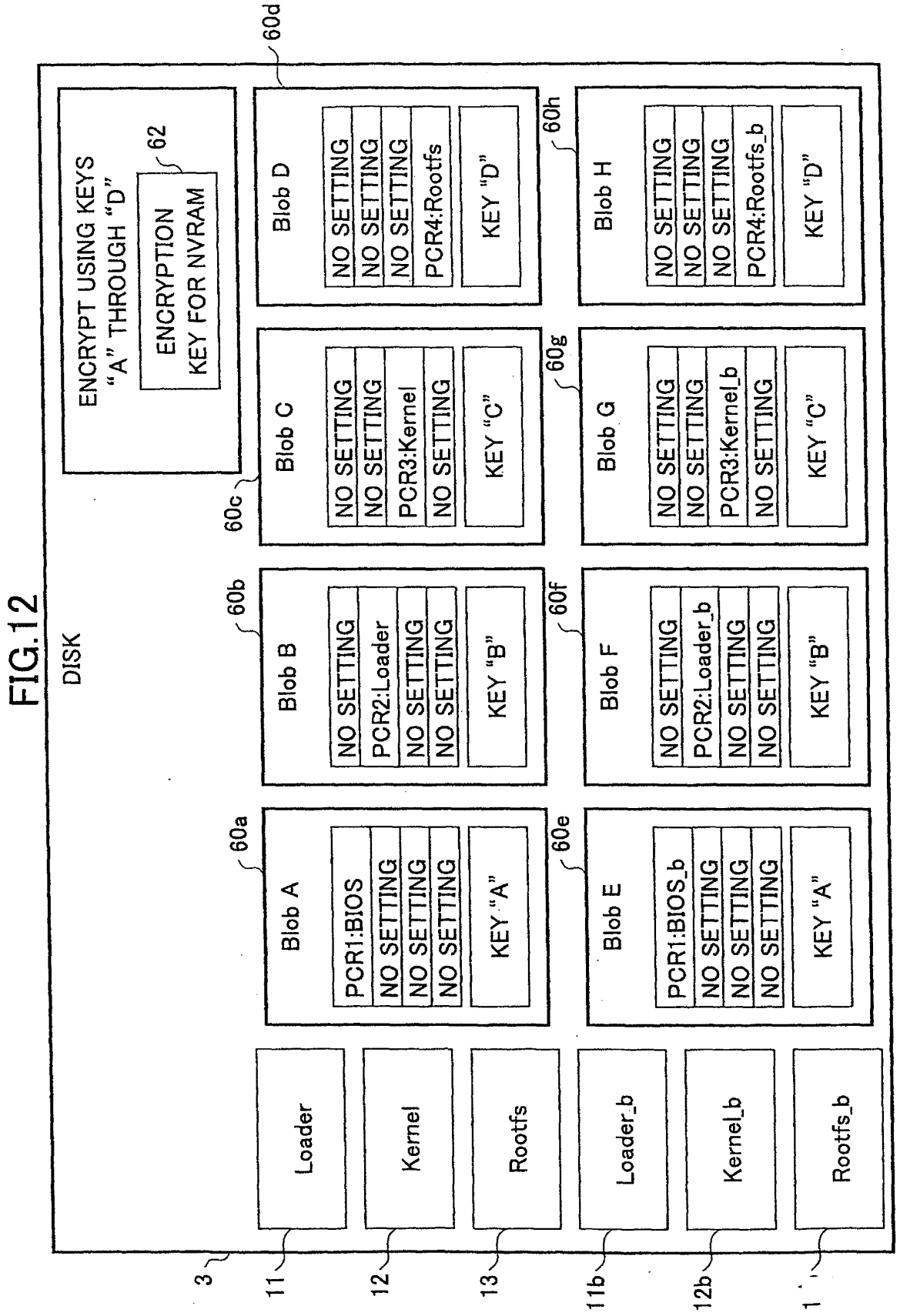

DISK
ENCRYPT USING KEYS "A" THROUGH "D"
ENCRYPTION KEY FOR NVRAM
62
60a
Blob A
PCR1:BIOS
NO SETTING
NO SETTING
NO SETTING
KEY "A"
60b
Blob B
NO SETTING
PCR2:Loader
NO SETTING
NO SETTING
KEY "B"
60c
Blob C
NO SETTING
NO SETTING
PCR3:Kernel
NO SETTING
KEY "C"
60d
Blob D
NO SETTING
NO SETTING
NO SETTING
PCR4:Rootfs
KEY "D"
60e
Blob E
PCR1:BIOS_b
NO SETTING
NO SETTING
NO SETTING
KEY "A"
60f
Blob F
NO SETTING
PCR2:Loader_b
NO SETTING
NO SETTING
KEY "B"
60g
Blob G
NO SETTING
NO SETTING
PCR3:Kernel_b
NO SETTING
KEY "C"
60h
Blob H
NO SETTING
NO SETTING
NO SETTING
PCR4:Rootfs_b
KEY "D"
Loader
Kernel
Rootfs
Loader_b
Kernel_b
Rootfs_b
3
11
12
13
11b
12b
1

FIG.13

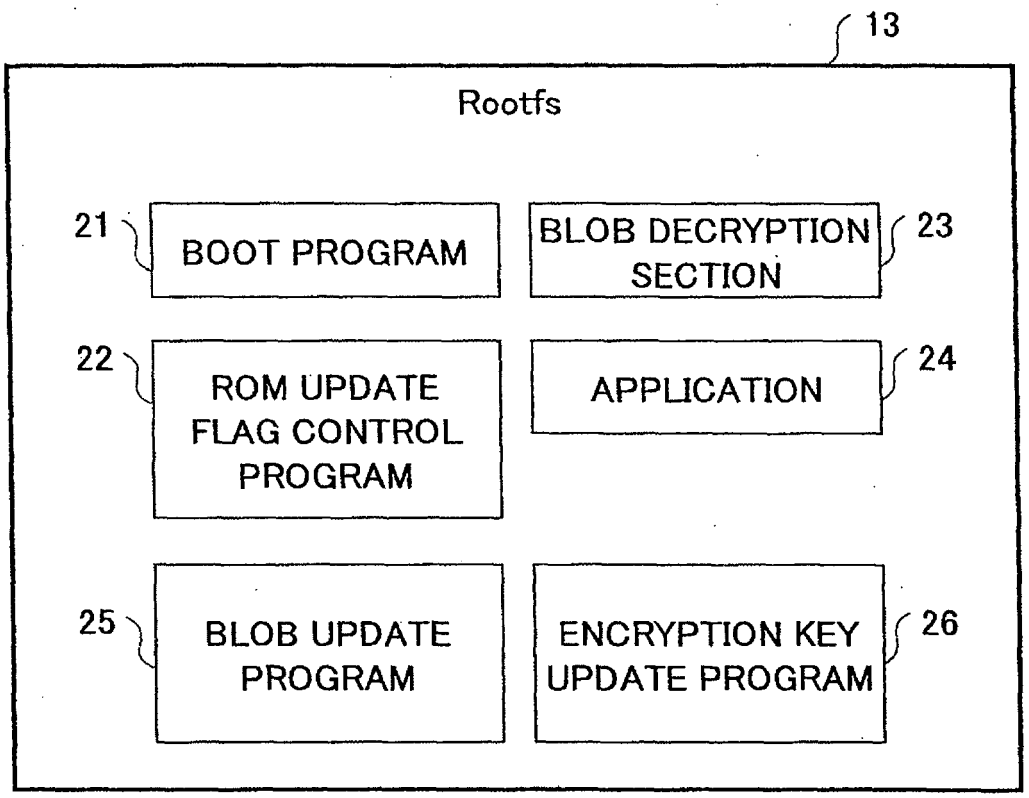

13
Rootfs
21
BOOT PROGRAM
BLOB DECRYPTION SECTION
23
22
ROM UPDATE FLAG CONTROL PROGRAM
APPLICATION
24
25
BLOB UPDATE PROGRAM
ENCRYPTION KEY UPDATE PROGRAM
26

FIG.14

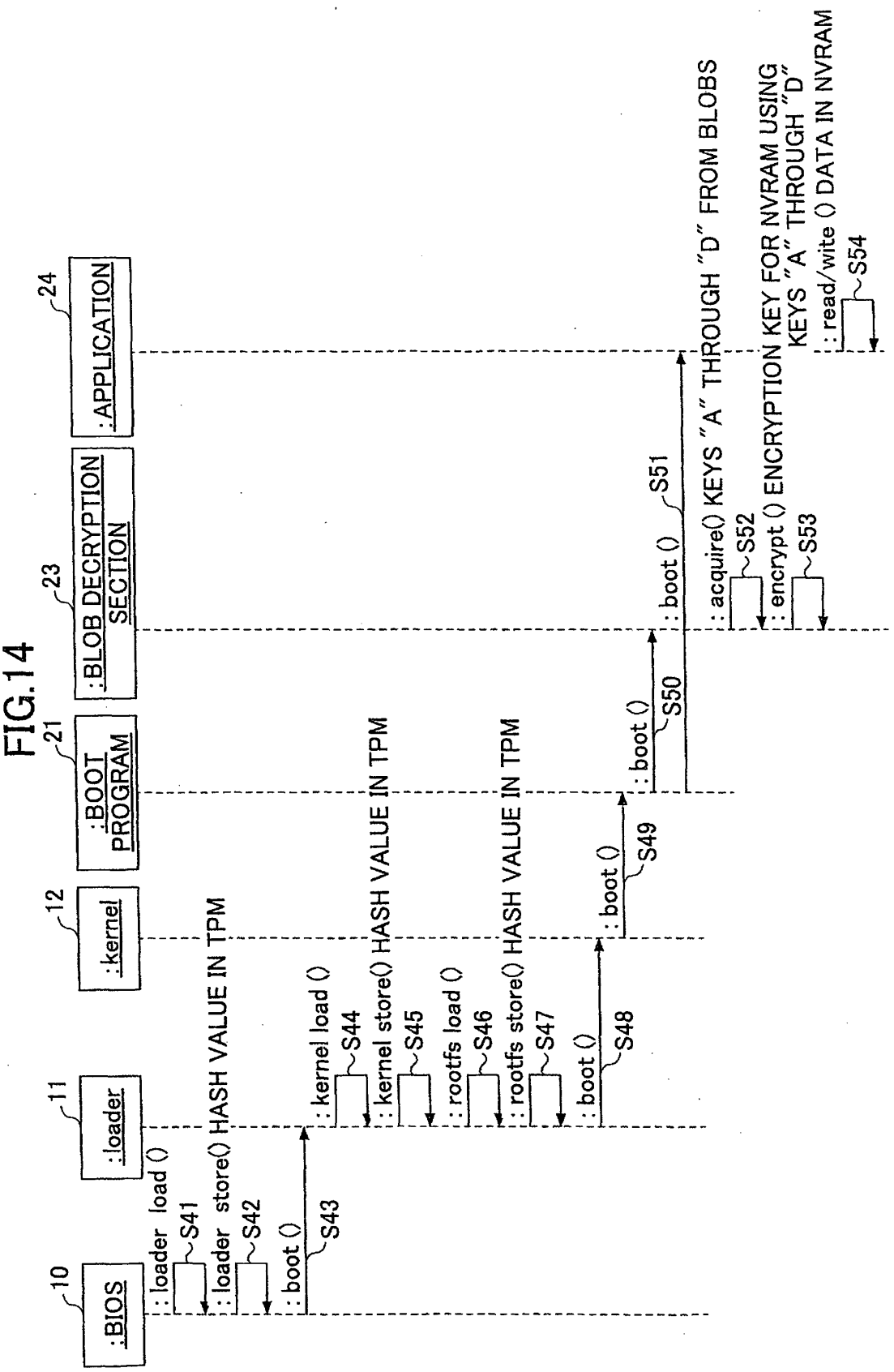

10
:BIOS
11
:loader
12
:kernel
21
:BOOT PROGRAM
23
:BLOB DECRYPTION SECTION
24
:APPLICATION
: loader load ()
S41
: loader store() HASH VALUE IN TPM
S42
: boot ()
S43
: kernel load ()
S44
: kernel store() HASH VALUE IN TPM
S45
: rootfs load ()
S46
: rootfs store() HASH VALUE IN TPM
S47
: boot ()
S48
: boot ()
S49
: boot ()
S50
: boot ()
S51
: acquire() KEYS "A" THROUGH "D" FROM BLOBS
S52
: encrypt () ENCRYPTION KEY FOR NVRAM USING KEYS "A" THROUGH "D"
S53
: read/wite () DATA IN NVRAM
S54

FIG.15

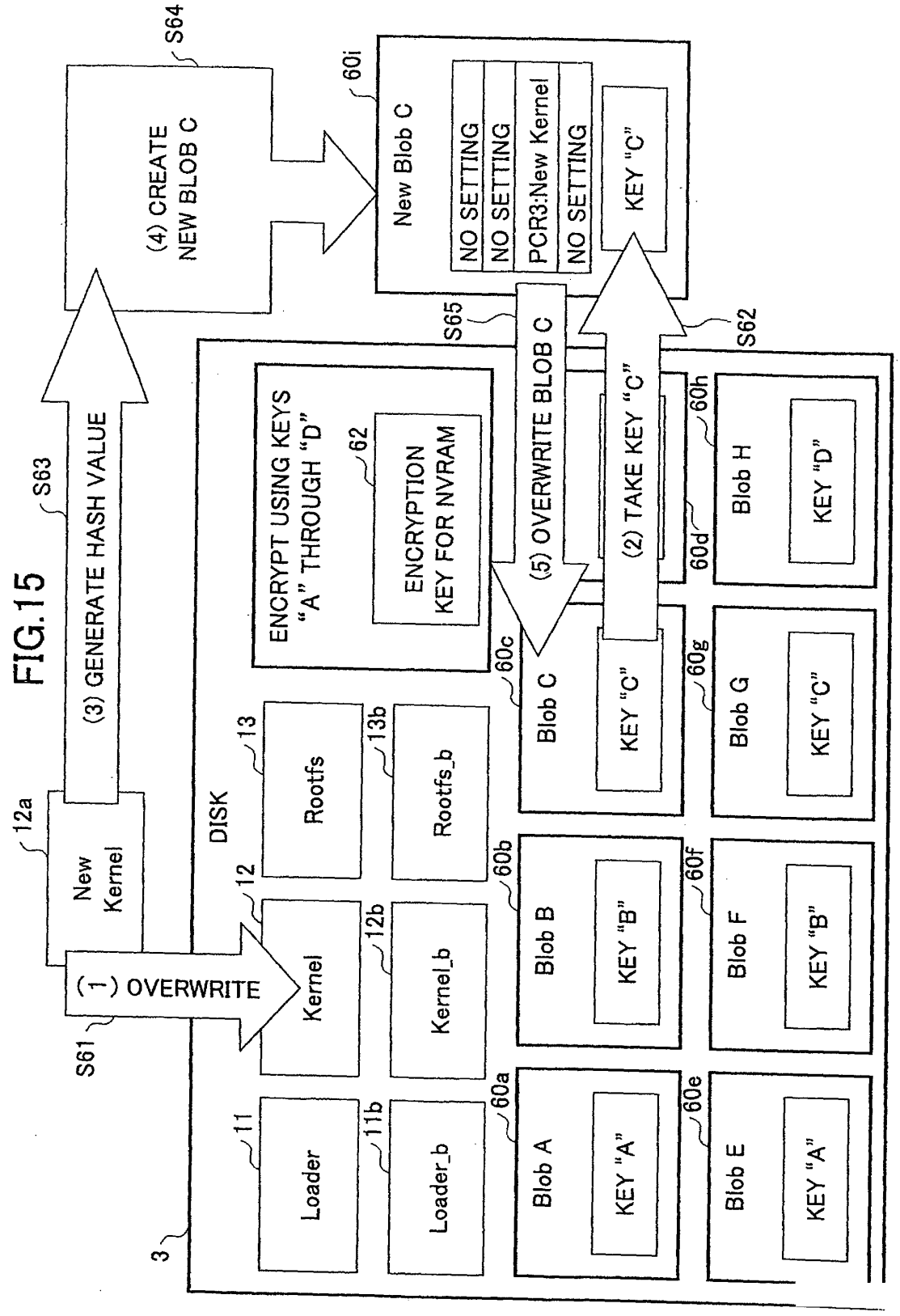

S64
(4) CREATE NEW BLOB C
60i
New Blob C
NO SETTING
NO SETTING
PCR3:New Kernel
NO SETTING
KEY "C"
S63
(3) GENERATE HASH VALUE
S65
(5) OVERWRITE BLOB C
S62
(2) TAKE KEY "C"
12a
New Kernel
DISK
ENCRYPT USING KEYS "A" THROUGH "D"
62
ENCRYPTION KEY FOR NVRAM
60h
Blob H
KEY "D"
60d
13
Rootfs
13b
Rootfs_b
60c
Blob C
KEY "C"
60g
Blob G
KEY "C"
(1) OVERWRITE
S61
12
Kernel
12b
Kernel_b
60b
Blob B
KEY "B"
60f
Blob F
KEY "B"
11
Loader
11b
Loader_b
60a
Blob A
KEY "A"
60e
Blob E
KEY "A"
3

FIG.16

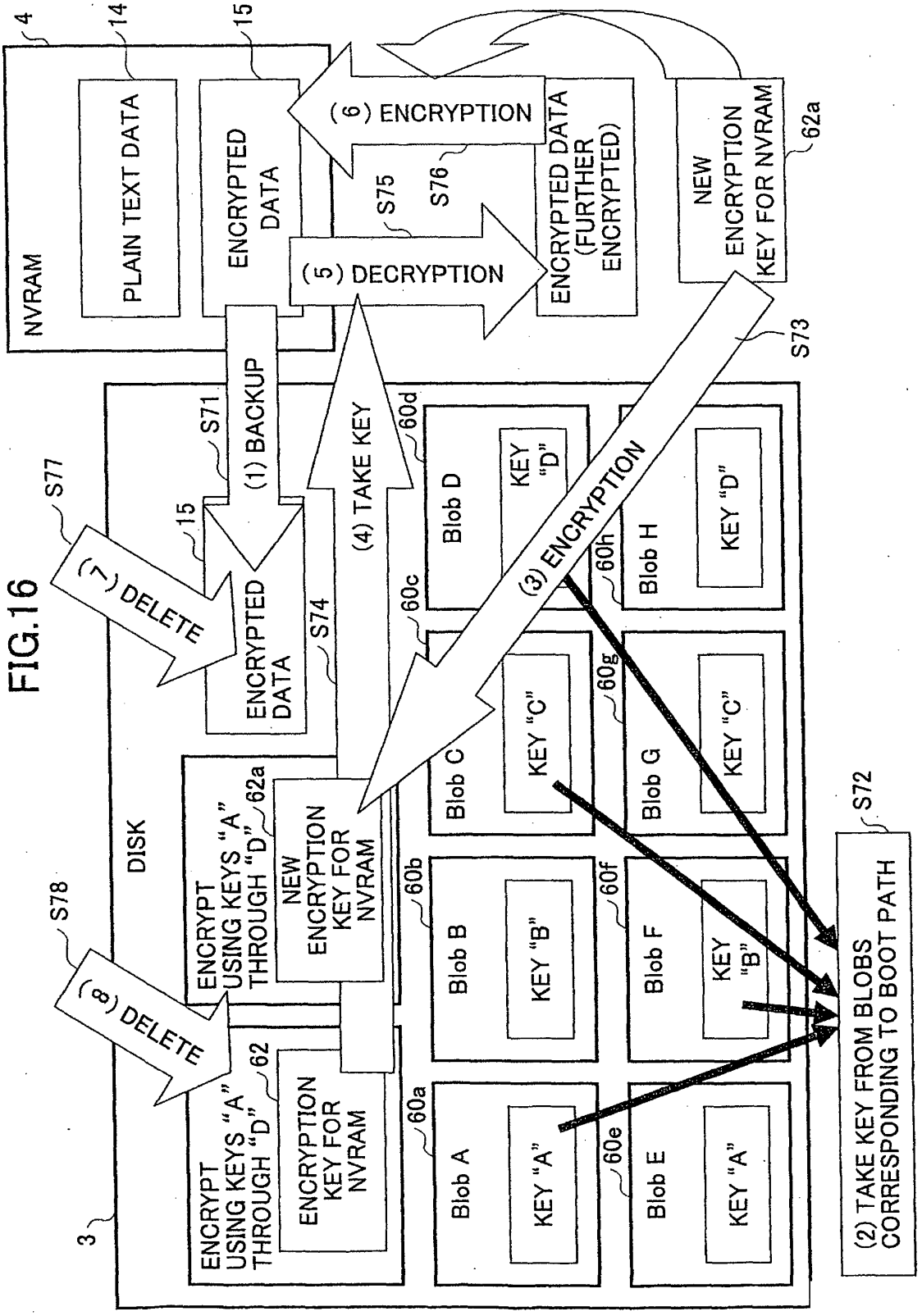

4
NVRAM
14
PLAIN TEXT DATA
15
ENCRYPTED DATA
(6) ENCRYPTION
S75
S76
ENCRYPTED DATA (FURTHER ENCRYPTED)
(5) DECRYPTION
NEW ENCRYPTION KEY FOR NVRAM
62a
S73
S71
(1) BACKUP
(4) TAKE KEY
60d
Blob D
KEY "D"
Blob H
KEY "D"
S77
(7) DELETE
15
ENCRYPTED DATA
S74
60c
(3) ENCRYPTION
60h
DISK
ENCRYPT USING KEYS "A" THROUGH "D"
62a
NEW ENCRYPTION KEY FOR NVRAM
Blob C
KEY "C"
60g
Blob G
KEY "C"
S72
S78
(8) DELETE
60b
Blob B
KEY "B"
60f
Blob F
KEY "B"
(2) TAKE KEY FROM BLOBS CORRESPONDING TO BOOT PATH
ENCRYPT USING KEYS "A" THROUGH "D"
62
ENCRYPTION KEY FOR NVRAM
60a
Blob A
KEY "A"
60e
Blob E
KEY "A"
3

FIG.17

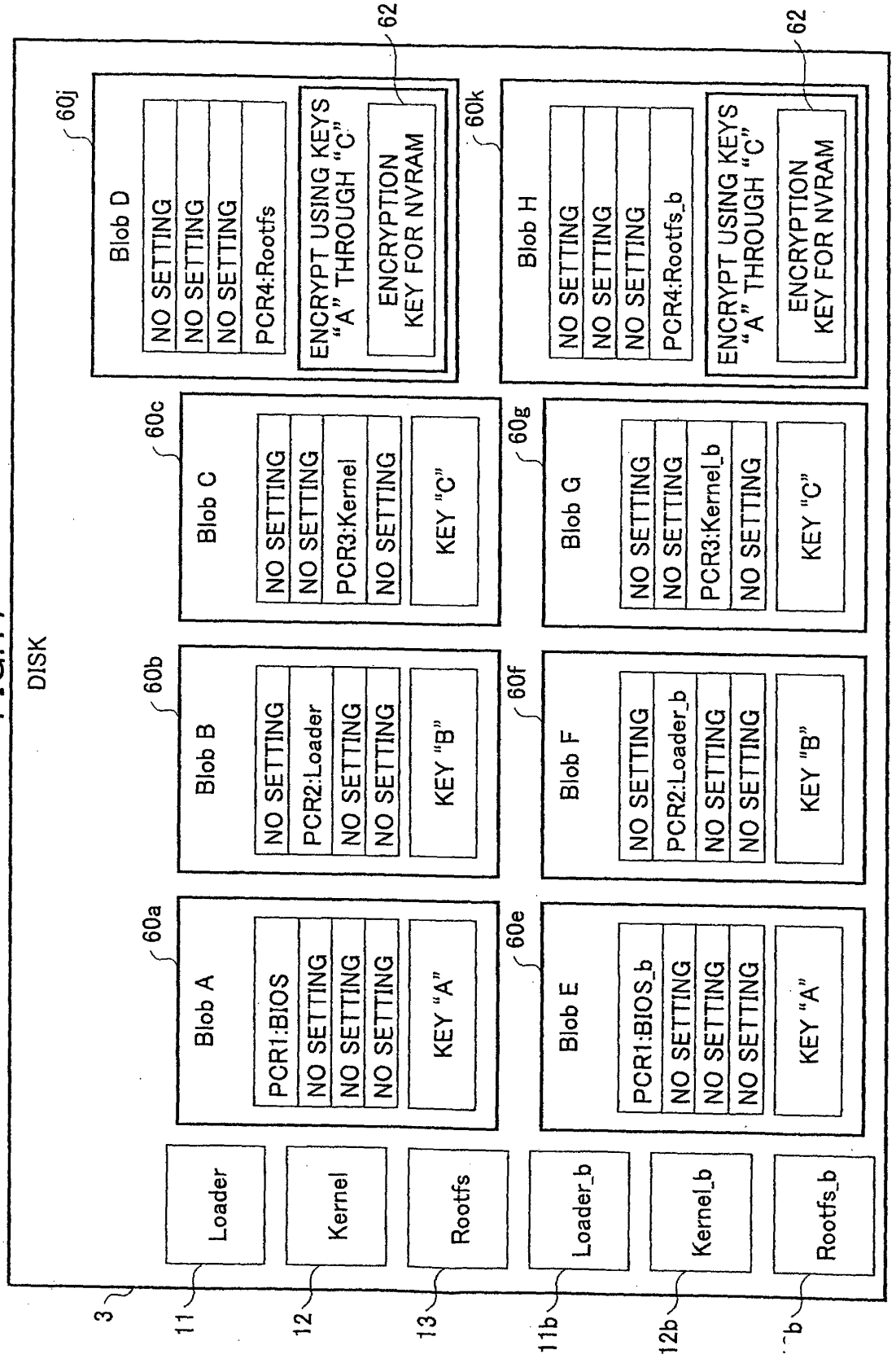

DISK
3
Loader
11
Kernel
12
Rootfs
13
Loader_b
11b
Kernel_b
12b
Rootfs_b
b
60a
Blob A
PCR1:BIOS
NO SETTING
NO SETTING
NO SETTING
KEY "A"
60b
Blob B
NO SETTING
PCR2:Loader
NO SETTING
NO SETTING
KEY "B"
60c
Blob C
NO SETTING
NO SETTING
PCR3:Kernel
NO SETTING
KEY "C"
60e
Blob E
PCR1:BIOS_b
NO SETTING
NO SETTING
NO SETTING
KEY "A"
60f
Blob F
NO SETTING
PCR2:Loader_b
NO SETTING
NO SETTING
KEY "B"
60g
Blob G
NO SETTING
NO SETTING
PCR3:Kernel_b
NO SETTING
KEY "C"
60j
Blob D
NO SETTING
NO SETTING
NO SETTING
PCR4:Rootfs
ENCRYPT USING KEYS "A" THROUGH "C"
ENCRYPTION KEY FOR NVRAM
62
60k
Blob H
NO SETTING
NO SETTING
NO SETTING
PCR4:Rootfs_b
ENCRYPT USING KEYS "A" THROUGH "C"
ENCRYPTION KEY FOR NVRAM
62

FIG.18

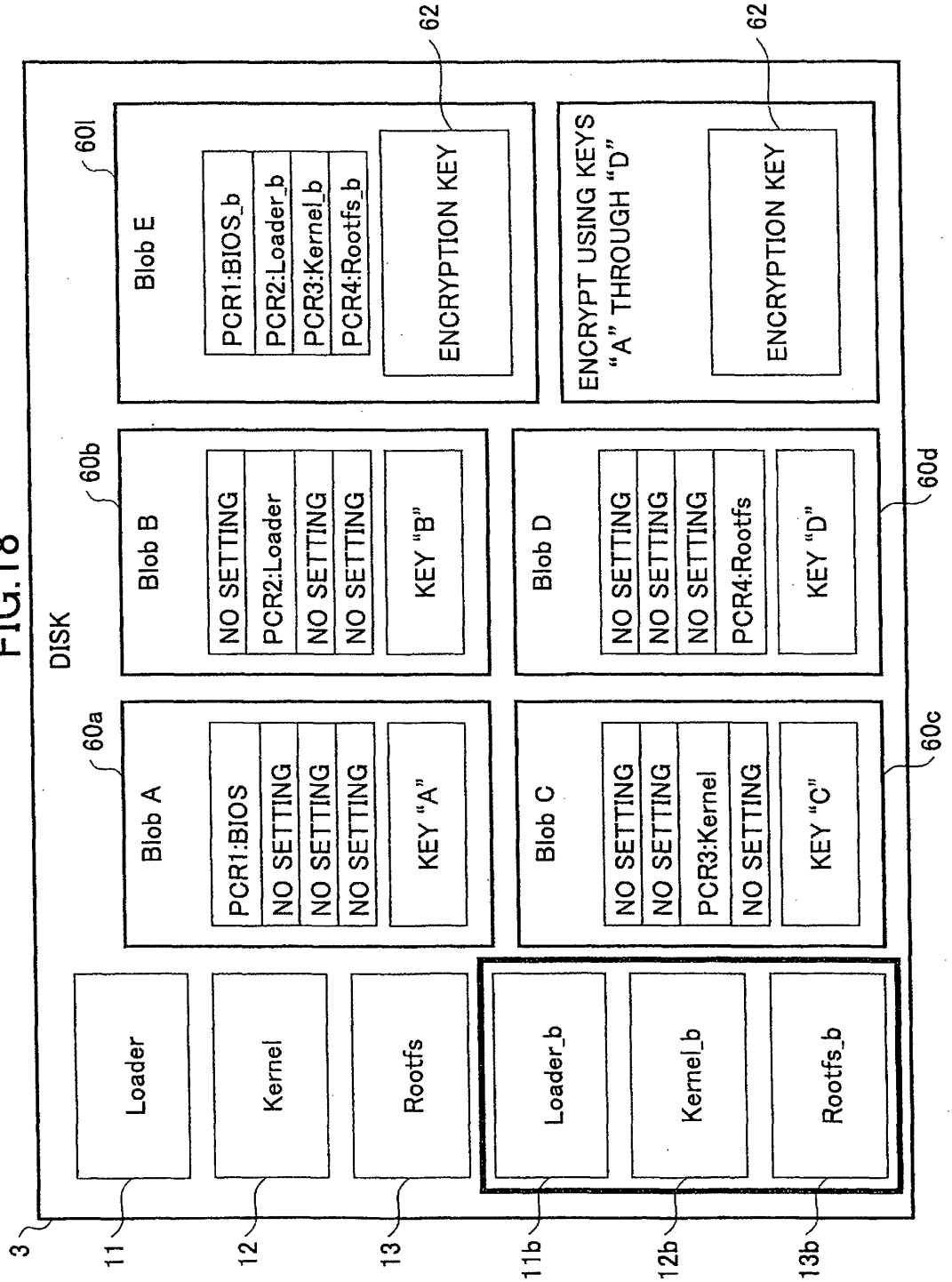

3
DISK
11
Loader
12
Kernel
13
Rootfs
11b
Loader_b
12b
Kernel_b
13b
Rootfs_b
60a
Blob A
PCR1:BIOS
NO SETTING
NO SETTING
NO SETTING
KEY "A"
60b
Blob B
NO SETTING
PCR2:Loader
NO SETTING
NO SETTING
KEY "B"
60c
Blob C
NO SETTING
NO SETTING
PCR3:Kernel
NO SETTING
KEY "C"
60d
Blob D
NO SETTING
NO SETTING
NO SETTING
PCR4:Rootfs
KEY "D"
60l
Blob E
PCR1:BIOS_b
PCR2:Loader_b
PCR3:Kernel_b
PCR4:Rootfs_b
ENCRYPTION KEY
62
ENCRYPT USING KEYS "A" THROUGH "D"
ENCRYPTION KEY
62

FIG.19

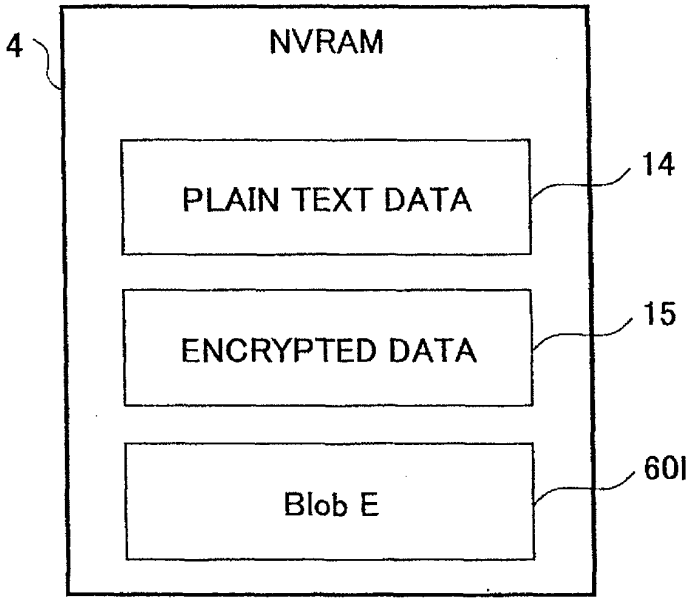

4
NVRAM
PLAIN TEXT DATA
14
ENCRYPTED DATA
15
Blob E
601

FIG.20

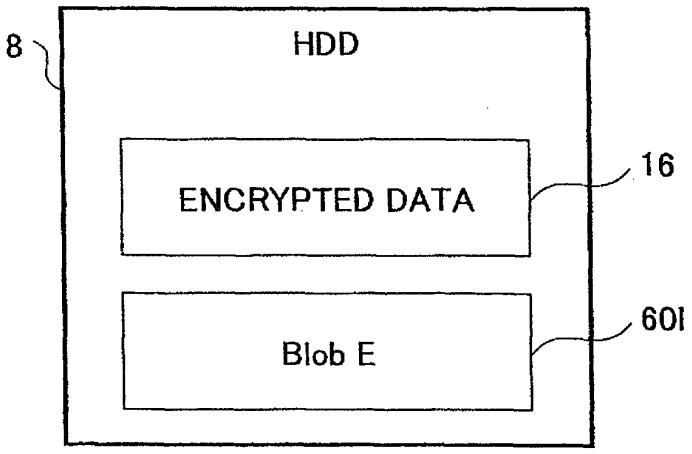

8
HDD
ENCRYPTED DATA
16
Blob E
601

FIG.21

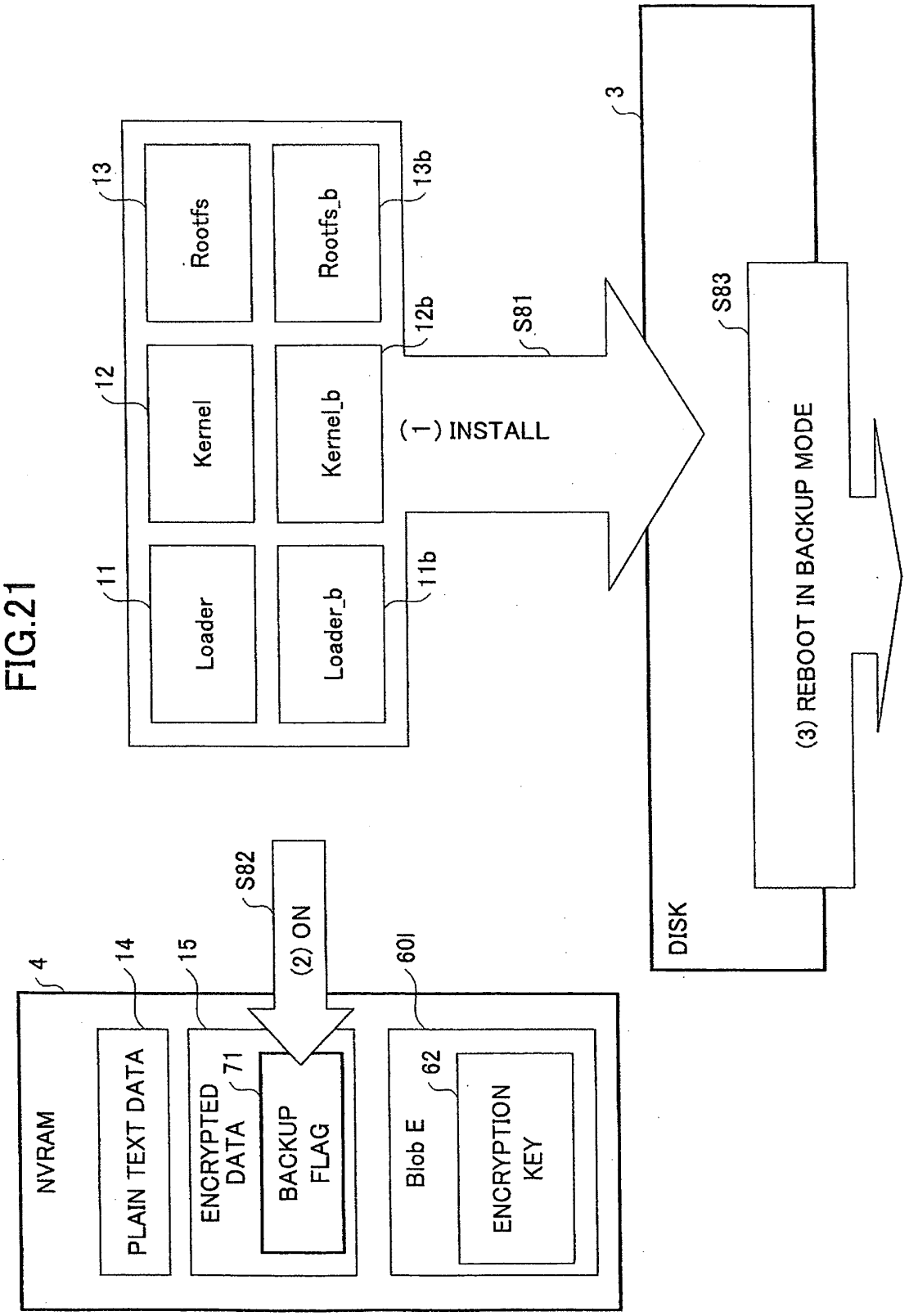
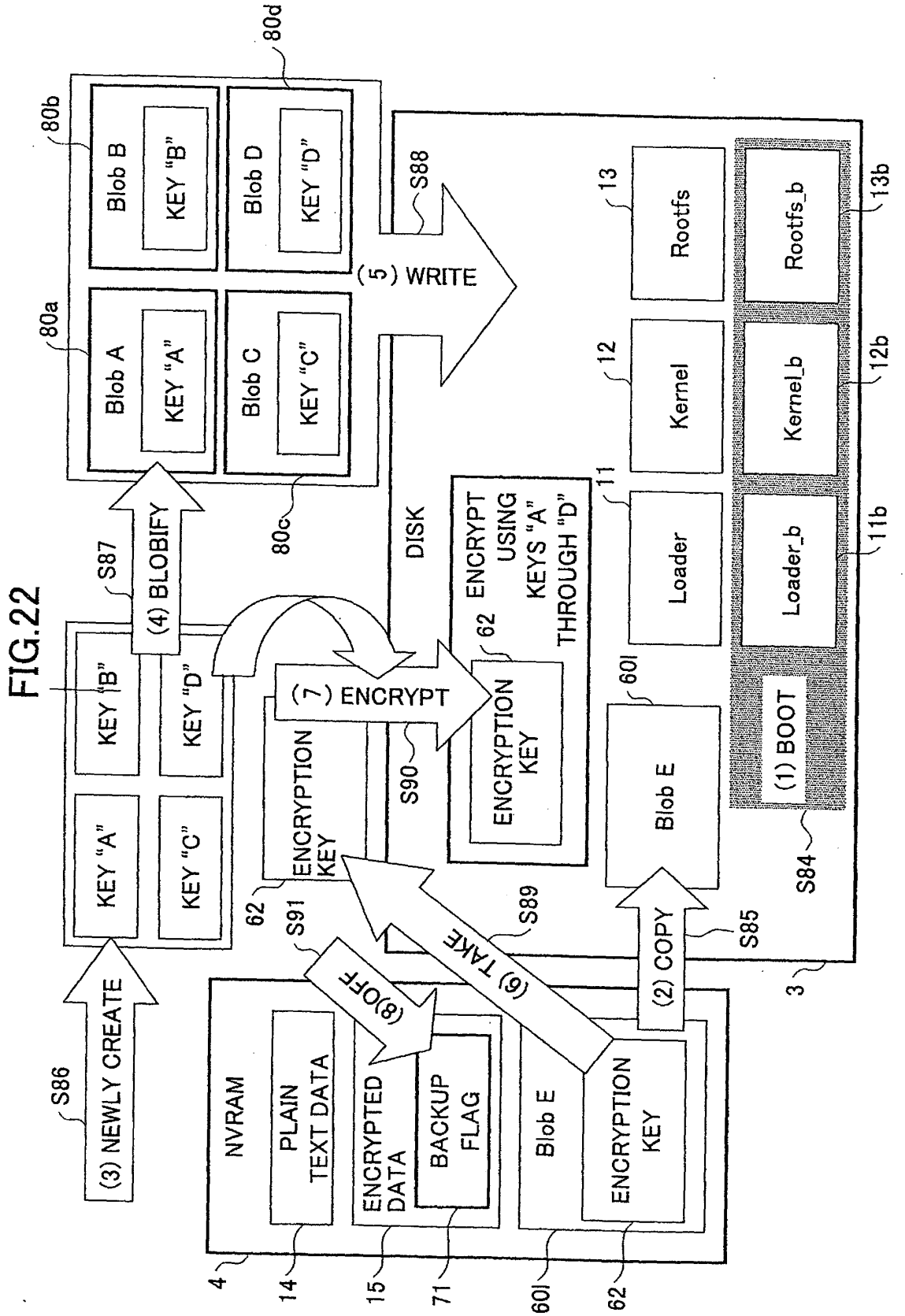

3
13
Rootfs
Rootfs_b
13b
12
Kernel
Kernel_b
12b
11
Loader
Loader_b
11b
S81
(1) INSTALL
S83
(3) REBOOT IN BACKUP MODE
DISK
S82
(2) ON
4
14
15
60l
NVRAM
PLAIN TEXT DATA
ENCRYPTED DATA
71
BACKUP FLAG
Blob E
62
ENCRYPTION KEY

FIG.22


80d
80b
Blob B
KEY "B"
Blob D
KEY "D"
S88
(5) WRITE
13
Rootfs
Rootfs_b
13b
80a
Blob A
KEY "A"
Blob C
KEY "C"
12
Kernel
Kernel_b
12b
80c
DISK
ENCRYPT USING KEYS "A" THROUGH "D"
11
Loader
Loader_b
11b
S87
(4) BLOBIFY
KEY "B"
KEY "D"
62
(7) ENCRYPT
ENCRYPTION KEY
601
Blob E
(1) BOOT
KEY "A"
KEY "C"
ENCRYPTION KEY
S90
S84
62
S91
S89
S85
(8)OFF
(6) TAKE
(2) COPY
3
S86
(3) NEWLY CREATE
NVRAM
PLAIN TEXT DATA
ENCRYPTED DATA
BACKUP FLAG
Blob E
ENCRYPTION KEY
4
14
15
71
601
62

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 6185678 B1 **[0004]**
- US 2003194094 A1 **[0005]**
- JP 2004282391 A **[0024]**
- JP 2005196745 A **[0025]**